MIT Open Access Articles

Folding equilateral plane graphs

Massachusetts Institute of Technology

Folding Equilateral Plane Graphs

Zachary Abel[1], Erik D. Demaine[2], Martin L. Demaine[2], Sarah Eisenstat[2],
Jayson Lynch[2], Tao B. Schardl[2], and Isaac Shapiro-Ellowitz[3]

[1] MIT Department of Mathematics, `zabel@math.mit.edu`
[2] MIT Computer Science and Artificial Intelligence Laboratory,
`{edemaine,mdemaine,seisenst,jaysonl,neboat}@mit.edu`
[3] University of Massachusetts Boston, `isaac.shapiroello001@umb.edu`

Abstract. 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, such reconfiguration is known to be impossi-
ble for linear (nonequilateral) trees and for (nonlinear) equilateral trees.
Second, under instantaneous folding motions, we show that an equilat-
eral plane graph has a noncrossing linear folded state if and only if it
is bipartite. Not only is the equilateral constraint necessary for this re-
sult, but we show 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 met-
ric polyhedral complex with one central vertex has a noncrossing flat
folded state with a specified "outside region". By contrast, the analogous
problem for a polyhedral manifold with one central vertex (*single-vertex
origami*) is only weakly NP-complete.

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. [2] 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 that the edges never properly cross each
other [3]. A tree is *universally foldable* [2] if it can be folded continuously from
any configuration to any other. 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, a tree is *locked*. The construction in [2] gives a specific tree

configuration that is *locked* in the sense that it cannot be folded to a canonical configuration.

Ballinger et al. [1] showed the existence of locked tree configurations with either of two special properties: (a) *linear*, where all edges lie along a line; and (b) *equilateral*, where all edge lengths are equal. The constructed examples of each type seem very different; for example, the locked equilateral tree has no touching bars (except at common endpoints), while the locked linear trees necessarily have many 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. 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 [8] describes precisely when a single-vertex crease pattern can fold flat using exactly the prescribed creases; see also [6, Thm. 12.2.1].

As described in [6, ch. 12], 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: see [6, Thm. 12.2.2] and [9].

Compared to locked trees, a key difference here is that we are interested only in *instantaneous* motions, and thus whether a linear configuration exists.

We consider the generalized problem of instantaneous folding of plane graphs (instead of just cycles) into linear configurations. Mapped to the context of origami, this problem is equivalent to flat folding of a *single-vertex complex*, consisting 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. This situation models the local behavior of a vertex neighborhood in a polyhedral complex (3D polygons attached at edges or vertices).

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: every cycle with a linear configuration must have an even number of edges, for the linear configuration naturally partitions the edges into two classes, one for each

direction along the line. 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. 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. Our reduction is from planar monotone 3-SAT, which was recently shown to be NP-complete [4].

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. 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 outside region must be specified; this technical requirement can likely be removed with additional effort. Second, the constructed complex might not be embeddable on the sphere with the given edge lengths. Third, 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) $\ell : E(G) \to \mathbb{R}_{>0}$ 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**, which can self-intersect, but cannot combinatorially cross itself. Connelly et al. [3] 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.*

In the next definition, a **chain** is a path of two unequal edges.

Definition 2. *For two weighted or unweighted plane graphs G and H, a **plane homomorphism** $g : G \to H$ is a graph homomorphism (preserving edge weights in the weighted case) together with, for each oriented edge $e \in E(H)$ from w_1*

to w_2, a linear (counterclockwise) ordering around w_1 of the edges $g^{-1}(e)$ (the set of 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 around w defined by g induce a cyclic order of all edges in G whose images are incident to w. The compatibility and planarity constraints may be expressed as follows:

- **Edge Ordering Compatibility**: For every oriented edge e in H, the linear orders for e and for e with opposite orientation are reversed, i.e., edges $g^{-1}(e)$ are linearly ordered.
- **Respect for Planar Embeddings**: For each vertex $v \in G$, the cyclic order of the edges incident to v around $g(v)$ induced by g matches their cyclic order around v in G.
- **Noncrossing**: For any two vertices v, 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.*

We interpret a plane homomorphism $g : G \to H$ as a "self touching embedding of G along graph H." More explicitly, for each vertex $w \in H$, the **magnified view** of g around w is 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 in the cyclic order 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.

The following lemmas, proven in the full version, make plane homomorphisms particularly useful:

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

Lemma 2. *A plane homomorphism $g : G \to H$ 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 $g^*(M)$.*

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 [1]. 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. [1] call this a "flat configuration"; we use the term "canonical configuration" 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 desired vertex v' and 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' [2].

Not all linear trees can be deformed into a canonical state; Ballinger et al. provide multiple such examples [1]. Likewise, not all equilateral trees are universally foldable: as shown in [], 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 $g : G \to H$. 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** *$\mathrm{thr}(e)$ to be the first edge in that ordering. Furthermore, if $\mathrm{thr}(e)$ 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 $g_k : G \to H_k$ such that the configuration of G is the one induced by g_k. Note that if a tree on n nodes is in an n-canonical state, then the tree is canonical.*

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 $2 \le \ell \le n$. We will show, by induction on $\ell \le k \le n$, 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 $g_k : G \to H_k$, where H_k is a tree on k vertices in a canonical state. Let $p_1, p_2, \ldots, p_{2k-2}$ be the vertices of the outer face of H_k in clockwise order around this face, and define $d_1 = \mathrm{thr}(p_1, p_2), d_2 =$
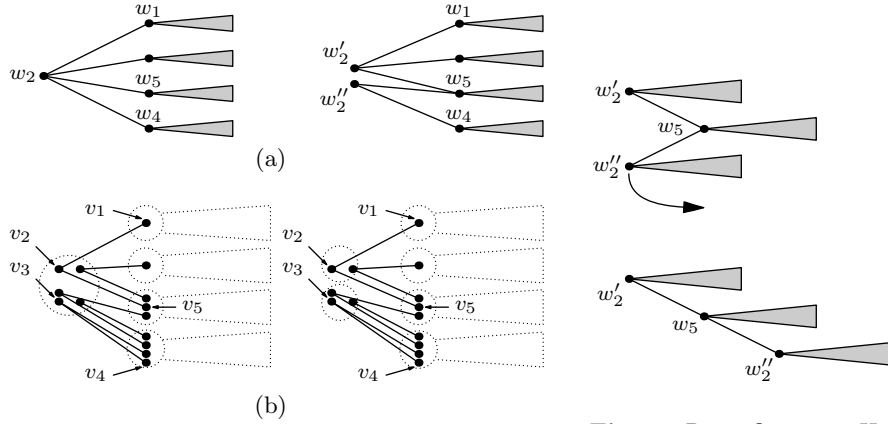
Fig. 1. An example of splitting a vertex. Figure (a) shows how the split is used to transform H_k into H_{k+1}. Figure (b) shows how the graph G is affected.

Fig. 2. Reconfiguring H_{k+1} into a canonical state.

$\text{thr}(p_2, p_3), \ldots, d_{2k-2} = \text{thr}(p_{2k-2}, p_1)$. 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 [2], 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 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, or both.

We next show how to "split" the vertex w_2 along edge (w_2, w_5) to construct a surjective plane graph homomorphism $g_{k+1} : G \to H_{k+1}$, 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 Fig. 1(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 $h : H_{k+1} \to H_k$ 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 $g_{k+1} : G \to H_{k+1}$ defined as follows: In the counterclockwise order 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 Fig. 1(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 Fig. 2, 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.   □

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 $G \to H_2$ will yield a linear state of G.

The base case $k = n$ is satisfied by the identity homomorphism $G \to G$. Now suppose we have a plane homomorphism $h_k : G \to H_k$ 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 3 edges. 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, $G \to H_k \to H_{k-1}$ proves the inductive step. Any configuration of H_2 must be linear, and therefore the configuration induced by $G \to H_2$ is also linear.   □

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 [7] by creating a cycle whose edge lengths are the numbers to partition. Hence, it is weakly NP-hard to determine whether a weighted graph has a linear state. In this section, we show that the problem is strongly NP-hard via a reduction from planar monotone 3-SAT, which is known to be NP-hard [4].

Let $G = (U \cup (C_+ \cup C_-), E)$ be a plane graph encoding an instance of the planar monotone 3-SAT problem. Specifically, let $U = (x_1, x_2, \ldots, x_n)$ 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 $\overline{x}$.

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 variables 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.

To perform this reduction, we represent each variable using an instance of the gadget shown in Fig. 3(a), and we represent each clause using an instance of the gadget shown in Fig. 3(c). We now discuss these gadgets in detail.

For each variable x_i with degree k in G, we construct a variable gadget with k entries, as exemplified in Fig. 3(a). These k entries correspond to the k copies of x_i in G'. On each side of the variable gadget, we consider two sets of points. We call the points $\{v_{i,0}, \ldots, v_{i,k}\}$ the *(positive) spine points* of the variable gadget and the points $\{w_{i,1}, \ldots, w_{i,k}\}$ the *(positive) flex points* of the variable gadget. Similarly, we define the points $\{v'_{i,0}, \ldots, v'_{i,k}\}$ and $\{w'_{i,0}, \ldots, w'_{i,k}\}$ to be the *(negative) spine points* and *(negative) flex points* respectively.[4] It can be shown that the resulting gadget has two linear states. The first state, illustrated in Fig. 3(a), has the flex points pointing right (i.e. positioned to the right of their spine points) and indicates $x_i = $ true, while the other has the flex points pointing left and indicates $x_i = $ false. We then connect the variable gadgets together to form the *variable column*, as shown in Fig. 3(b). This variable column has exactly 2^n linear states, corresponding to all possible boolean assignments to the variables U. Moreover, each of these linear states ensures that the edges between the spine and flex points have the same top-to-bottom ordering as Fig. 3(b).

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

---

[4] We omit the "positive" and "negative" specifiers when it is clear from context.

(a) Variable gadget   (b) Variable column   (c) Clause gadget
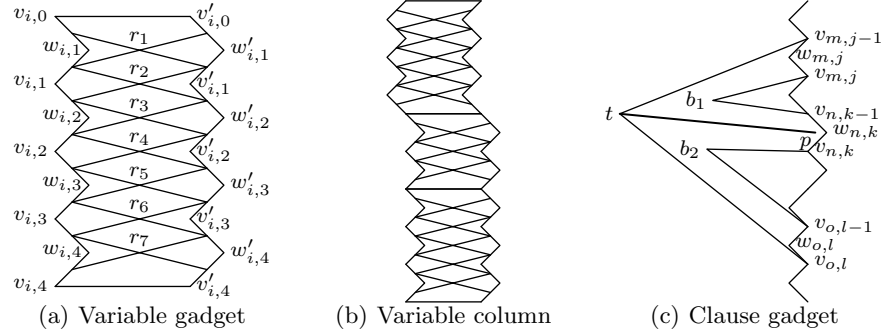
Fig. 3. Figure (a) shows a 4-entry variable gadget, set to true. The edge length between any pair of adjacent spine and truth points on one side is 2. Figure (b) shows an example variable column containing three variable gadgets. The bottom two variables represented are set to true, while the top-most 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.

vertices t, p, b_1, b_2 and six new edges as depicted in Fig. 3(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 interfering with each other. The **_probe_** (t, p) is a long 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 edges, prevent the probe from accessing variable entries for variables not in c. By construction, the clause gadget has the following property.

Lemma 3. *For each clause $c \in C_+ \cup C_-$, 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 Fig. 3(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 Fig. 3(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. $\square$

If G is an instance of planar monotone 3-SAT and H is the graph obtained by this reduction, then by Lemma 3 and preceeding discussion, linear configurations of H correspond exactly to satisfying assignments of G. We thus obtain:

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

References

1. 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*, volume 5664 of *Lecture Notes in Computer Science*, pages 61–73, Banff, Canada, August 2009.
2. 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(1–3):293–297, 2002.
3. R. Connelly, E. D. Demaine, and G. Rote. Infinitesimally locked self-touching linkages with applications to locked trees. In J. Calvo, K. Millett, and E. Rawdon, editors, *Physical Knots: Knotting, Linking, and Folding of Geometric Objects in R3*, pages 287—311. American Mathematical Society, 2002.
4. 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*, volume 6196 of *Lecture Notes in Computer Science*, pages 216–225, 2010.
5. 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*, pages 64–67, Montréal, Canada, August 2004.
6. E. D. Demaine and J. O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, July 2007.
7. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103, New York, NY, 1972. Plenum Press.
8. T. Kawasaki. On the relation between mountain-creases and valley-creases of a flat origami. In H. Huzita, editor, *Proceedings of the 1st International Meeting of Origami Science and Technology*, pages 229–237, Ferrara, Italy, December 1989. An unabridged Japanese version appeared in *Sasebo College of Technology Report*, 27:153–157, 1990.
9. 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*, volume 3742 of *Lecture Notes in Computer Science*, pages 161–173, Tokyo, Japan, October 2004.