Contents

A Survey of Folding and Unfolding in Computational Geometry

Erik D. Demaine* Joseph O'Rourke†

Abstract

We survey results in a recent branch of computational geometry: folding and unfolding of linkages, paper, and polyhedra.

1 Introduction

Folding and unfolding problems have been implicit since Albrecht Dürer in the early 1500's [Dür77], but have not been studied extensively in the mathematical literature until recently. Over the past few years, there has been a surge of interest in these problems in discrete and computational geometry. This paper gives a brief survey of most of the work in this area. Related, shorter surveys are given by Connelly and Demaine [CD04], Demaine [Dem00], Demaine and Demaine [DD01], and O'Rourke [O'R98]. We are currently preparing a monograph on the topic [DO].

In general, we are interested in how objects (such as linkages, pieces of paper, and polyhedra) can be moved or reconfigured (folded) subject to certain constraints depending on the type of object and the problem of interest. Typically the process of *unfolding* approaches a more basic shape, whereas *folding* complicates the shape. We define the *configuration space* as the set of all configurations or states of the object permitted by the folding constraints, with paths in the space corresponding to motions (foldings) of the object.

This survey is divided into three sections corresponding to the type of object being folded: linkages, paper, or polyhedra. Unavoidably, areas with which we are more familiar or for which there is a more extensive literature are covered in more detail. For example, more problems have been explored in linkage and paper folding than in polyhedron folding, and our corresponding sections reflect this imbalance. On the other hand, this survey cannot do justice to the wealth of research on protein folding, so only a partial survey appears in Section 2.5.

*MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139, USA, edemaine@mit.edu. Supported by NSF CAREER award CCF-0347776.

†Dept. Comput. Sci., Smith College, Northampton, MA 01063, USA, orourke@cs.smith.edu. Supported by NSF Distinguished Teaching Scholars award DUE-0123154.

2 Linkages

2.1 Definitions and Fundamental Questions

A *linkage* or *framework* consists of a collection of rigid line segments (*bars* or *links*) joined at their endpoints (*vertices* or *joints*) to form a particular graph. A linkage can be *folded* by moving the vertices in $\mathbb{R}^d$ in any way that preserves the length of every bar. Unless otherwise specified, we assume the vertices to be universal joints, permitting the full angular range of motions. Restricted angular motions will be discussed in Section 2.5.2.

Linkages have been studied extensively in the case that bars are permitted to cross; see, for example, [HJW84, JS99, KM95, Kem76, LW95, Sal73, Whi92]. Such linkages can be very complex, even in the plane. In 1876, Kempe [Kem76] suggested an incomplete argument to show that a planar linkage can be built so that a vertex traces an arbitrary polynomial curve—there is a linkage that can "sign your name." It was not until recently that Kempe's claim was established rigorously by Kapovich and Millson [KM02]. Hopcroft, Joseph, and Whitesides [HJW84] showed that deciding whether a planar linkage can reach a particular configuration is PSPACE-complete. Jordan and Steiner [JS99] proved that there is a linkage whose configuration space is homeomorphic to an arbitrary compact real algebraic variety with Euclidean topology, and thus planar linkages are equivalent to the theory of the reals (solving systems of polynomial inequalities over reals). On the other hand, for a linkage whose graph is just a cycle, all configurations can be reached in Euclidean space of any dimension greater than 2 by a sequence of simple motions [LW95, Sal73], and in the plane there is a simple restriction characterizing which polygons can be inverted in orientation [LW95].

Recently there has been much work on the case that the linkage must remain *simple*, that is, never have two bars cross.[1] The remainder of this survey assumes this noncrossing constraint. Such linkage folding has applications in hydraulic tube bending [O'R98] and motion planning of robot arms. There are also connections to protein folding in molecular biology, which we touch upon in Section 2.5. See also [CDR03, O'R98, Tou99a] for other surveys on linkage folding without crossings.

Perhaps the most fundamental question one can ask about folding linkages is whether it is possible to fold between any two configurations. That is, is there a folding between any two simple configurations of the same linkage (with matching graphs, combinatorial embeddings, and bar lengths) while preserving the bar lengths and not crossing any bars during the folding? Because folding motions can be reversed and concatenated, this fundamental question is equivalent to whether every simple configuration can be folded into some *canonical configuration*, a configuration whose definition depends on the type of linkage under consideration.

We concentrate here on allowing all continuous motions that maintain simplicity, but we should mention that different applications often further constrain

[1] Typically, bars are allowed to touch, provided they do not properly intersect. However, insisting that bars only touch at common endpoints does not change the results.

the permissible motions in various ways. For example, hydraulic tube bending allows only one joint to bend at any one time, and moreover the joint angle can never reverse direction. Such constraints often drastically alter what is possible. See, e.g., [AFM03].

In the context of linkages whose edges cannot cross, three general types of linkages are commonly studied, characterized by the structure of their associated graphs (see Figure 1): a *polygonal arc* or *open polygonal chain* (a single path); a *polygonal cycle*, *polygon*, or *closed polygonal chain* (a single cycle); and a *polygonal tree* (a single tree).[2] The canonical configuration of an arc is the *straight configuration*, all vertex angles equal to 180°. A canonical configuration of a cycle is a *convex configuration*, planar and having all interior vertex angles less than or equal to 180°. It is relatively easy to show that convex configurations are indeed "canonical" in the sense that any one can be folded into any other, a result that first appeared in [ADE+01]. Finally, a canonical configuration of a tree is a *flat configuration*: all vertices lie on a horizontal line, and all bars point "rightward" from a common root. Again it is easy to fold any flat configuration into any other [BDD+02].

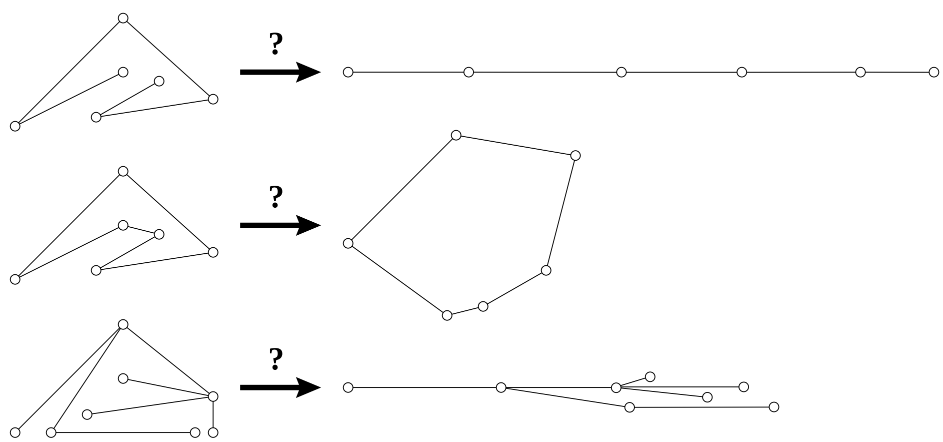


Figure 1: The three common types of linkages and their associated canonical configurations. From top to bottom, a polygonal arc $\overset{?}{\rightarrow}$ the straight configuration, a polygonal cycle $\overset{?}{\rightarrow}$ a convex configuration, and a polygonal tree $\overset{?}{\rightarrow}$ a (nearly) flat configuration.

The fundamental questions thus become whether every arc can be straightened, every cycle can be convexified, and every tree can be flattened. The answers to these questions depend on the dimension of the space in which the linkage starts, and the dimension of the space in which the linkage may be

[2]More general graphs have been studied largely in the context of allowing bars to cross, exploring either aspects of the configurations space (e.g., the Kempe work mentioned earlier), or the conditions which render the graph rigid. Graph rigidity is a rich topic, not detailed here, which also plays a role in the noncrossing-bar scenario in Section 2.2.1.

folded. Over the past few years, this collection of questions has been completely resolved. Table 1 summarizes the answers to these questions when the dimensions of the initial configuration and the folding process are equal. The answers in the columns for arcs and cycles are analogous to the existence of knots tied from one-dimensional string: nontrivial knots exist only in 3D. In contrast, the column for trees in Table 1 presents an interesting difference in 2D: while trees in the plane are topologically unknotted, they can be geometrically locked. This observation is some evidence for the belief that the fundamental problems are most difficult in 2D.

Dimension	Can all arcs be straightened?	Can all cycles be convexified?	Can all trees be flattened?
2	Yes [CDR03]	Yes [CDR03]	No [BDD+02]
3	No [CJ98, BDD+01]	No [CJ98, BDD+01]	No [from arcs]
4 & above	Yes [CO01]	Yes [CO01]	Yes [CO01]

Table 1: Answers to fundamental questions for linkage folding without crossings.

The next three subsections describe the historical progress of these results and other results closely related to the fundamental questions, one section per row in Table 1. Along the way, Sections 2.3.1–2.3.4 describe several special forms of linkage folding arising out of a problem posed by Erdős in 1935; and Section 2.3.8 considers the generalization of multiple chains. Finally, Section 2.5 discusses the connections between linkage folding and protein folding, and describes the most closely related results and open problems.

2.2 Fundamental Questions in 2D

Section 2.2.1 describes the development of the theorems for straightening arcs and convexifying cycles in 2D. Section 2.2.2 discusses the contrary result that not all trees can be flattened.

2.2.1 Carpenter's Rule Problem: Polygonal Chains in 2D

The questions of whether every polygonal arc can be straightened and every polygonal cycle can be convexified in the plane have arisen in many contexts over the last quarter-century.[3] In the discrete and computational geometry community, the arc-straightening problem has become known as the *carpenter's rule problem* because a carpenter's rule folds like a polygonal arc.

Most people's initial intuition is that the answers to these problems are YES, but describing a precise general motion proved difficult. It was not until 2000 that the problems were solved by Connelly, Demaine, and Rote [CDR03], with

[3] Posed independently by Stephen Schanuel and George Bergman in the early 1970's, Ulf Grenander in 1987, William Lenhart and Sue Whitesides in 1991, and Joseph Mitchell in 1992 [CDR03].

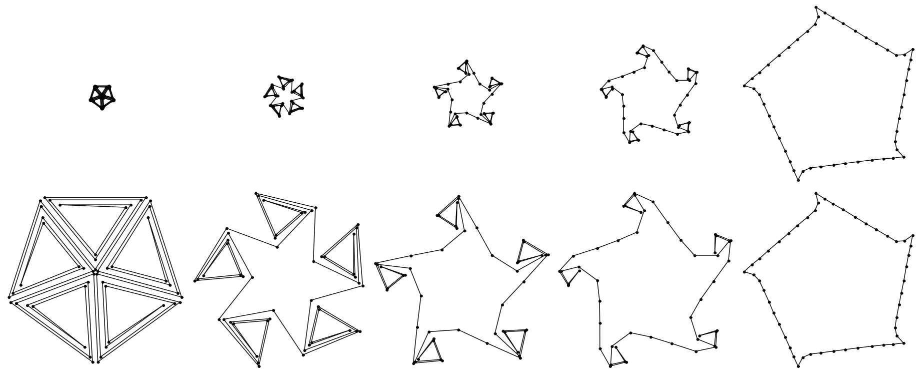


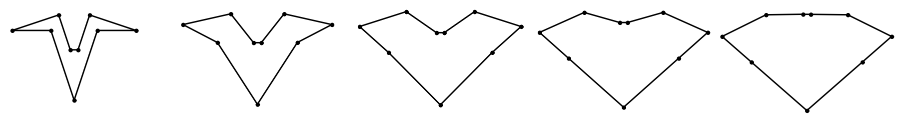
Figure 2: Two views of convexifying a "doubled tree" linkage. The top snapshots are all scaled the same, and the bottom snapshots are rescaled to improve visibility.

an answer of YES. Figure 2 shows an example of the motion resulting from this theorem.

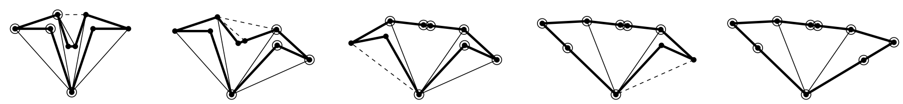
More generally, the result in [CDR03] shows that a collection of nonintersecting polygonal arcs and cycles in the plane may be simultaneously folded so that the outermost arcs are straightened and the outermost cycles are convexified. The "outermost" proviso is necessary because arcs and cycles cannot always be straightened and convexified when they are contained in other cycles. The key idea for the solution, introduced by Günter Rote, is to look for *expansive* motions in which no vertex-to-vertex distance decreases. Bars cannot cross before getting closer, so expansiveness allows us to ignore the difficult nonlocal constraint that bars must not cross. Expansiveness brings the problem into the areas of rigidity theory and tensegrity theory, which study frameworks of rigid bars, unshrinkable *struts*, and unexpandable *cables*. Tools from these areas helped show that, *infinitesimally*, arcs and cycles can be unfolded expansively. These infinitesimal motions are combined by flowing along a vector field defined implicitly by an optimization problem. As a result, the motion is piecewise-differentiable (C^1). In addition, any symmetries present in the initial configuration of the linkage are preserved throughout the motion. Similar techniques show that the area of each cycle increases by this motion and furthermore by any expansive motion [CDR03].

Since the original theorem, two additional algorithms have been developed for unfolding polygonal chains. Figure 3 provides a visual comparison of all three algorithms.

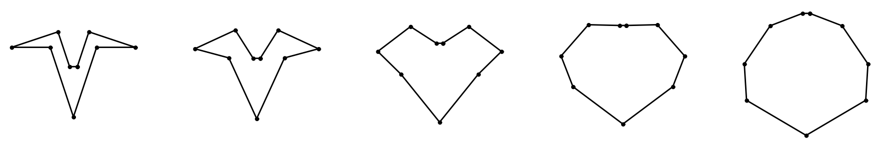
Ileana Streinu [Str00] demonstrated another expansive motion for straightening arcs and convexifying polygons that is piecewise-algebraic, composed of a polynomial-length sequence of *mechanisms*, each with a single degree of freedom. In this sense the motion is easier to implement "mechanically." It is also possible to compute the algebraic curves involved, though the running time is exponen-



(a) Via convex programming [CDR03]



(b) Via pseudotriangulations [Str00]. Pinned vertices are circled.



(c) Via energy minimization [CDIO04].

Figure 3: Convexifying a common polygon via all three convexification methods.

tial in n. This method also elucidates an interesting combinatorial structure to 2D linkage unfolding through "pseudotriangulations," which have subsequently received much attention in computational geometry (see, e.g., [O'R02, Rot03]).

Cantarella, Demaine, Iben, and O'Brien [CDIO04] gave an energy-based algorithm for straightening arcs and convexifying polygons. This algorithm follows the downhill gradient of an appropriate energy function, corresponding roughly to the intuition of filling the polygon with air. The resulting motion is not expansive, essentially averaging out the strut constraints. On the other hand, the existence of the downhill gradient relies on the existence of expansive motions from [CDR03], by showing that the latter decrease energy. The motion avoids self-intersection not through expansiveness but by designing the energy function to approach $+\infty$ near an intersecting configuration; any downhill flow avoids such spikes. The result is a C^∞ motion, easily computed as a piecewise-linear motion in angle space. The number of steps in the piecewise-linear motion is polynomial in two quantities: in the number of vertices n, and in the ratio between the maximum edge length and the initial minimum distance between a vertex and an edge.

2.2.2 Trees in 2D

In 1998, Biedl et al. [BDD$^+$02] showed that not all trees can be flattened in the plane. Their example consists of at least 5 *petals* connected at a central high-degree vertex. The version shown in Figure 4 uses 8 petals. Each petal is an arc of three bars, the last of which is "wedged" into the center vertex.

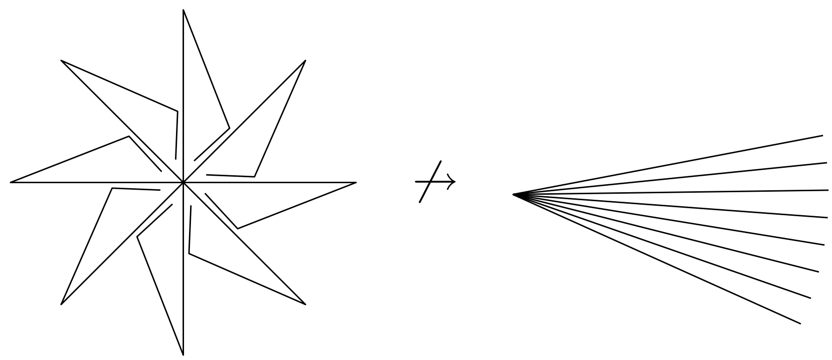


Figure 4: The locked tree on the left, from [BDD$^+$02], cannot be reconfigured into the nearly flat configuration on the right. (Figure 1 of [BDD$^+$02].)

Intuitively, the argument that the tree is locked is as follows. No petal can be straightened unless enough angular room has been made. But no petal can be reduced to occupy less angular space by more than a small positive number unless the petal has already been straightened. This circular dependence implies that no petal can be straightened, so the tree is locked. The details of this argument, in particular obtaining suitable tolerances for closeness, are somewhat intricate [BDD$^+$02]. The key is that each petal occupies a wedge of space whose angle is less than 90°, which is why at least 5 petals are required.

This tree remains locked if we replace the central degree-5 (or higher) vertex with multiple degree-3 vertices connected by very short bars [BDD$^+$02, full version]. Connelly, Demaine, and Rote [CDR02] showed that the tree in Figure 5, with a single degree-3 vertex and the remaining vertices having degrees 1 and 2, is locked, proving tightness of the arc-and-cycle result in [CDR03]. In [CDR02], they also present an extension to rigidity/tensegrity theory that permits establishing via linear programming that many classes of planar linkages (e.g., trees) are locked. In particular, they use this method to give short proofs that the tree in Figure 4 and the tree with one degree-3 vertex are *strongly locked*, in the sense that sufficiently small perturbations of the vertex positions and bar lengths result in a tree that cannot be moved more than ε in the configuration space for any $\varepsilon > 0$.

2.3 Fundamental Questions in 3D

Linkage folding in 3D was initiated earlier, by Paul Erdős in 1935 [Erd35]. His problem and its solution are described in Section 2.3.1. Sections 2.3.2–2.3.4

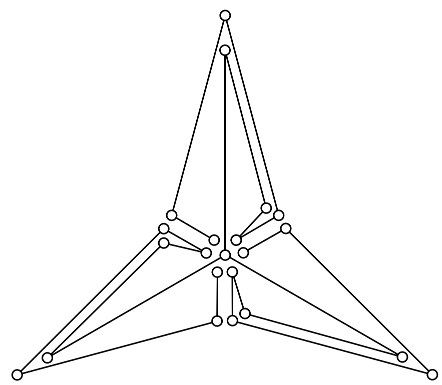


Figure 5: The locked tree from [CDR02]. (Based on Figure 1(c) of [CDR02].)

consider various extensions of this problem. All of this work deals with linkages that start in the plane, but fold through 3D. The more general situation, an arbitrary linkage starting in 3D, is addressed in Section 2.3.6. As this problem proves unsolvable in general, additional special cases are addressed in Section 2.3.7. Finally, Section 2.3.8 considers the generalized problem of multiple interlocking chains.

2.3.1 Flips for Planar Polygons in 3D

The roots of linkage folding go back to a problem Erdős posed in 1935 in the *American Mathematics Monthy* [Erd35]. Define a *pocket* of a polygon to be a region bounded by a subchain of the polygon edges, and define the *lid* of the pocket to be the edge of the convex hull connecting the endpoints of that subchain. Every nonconvex polygon has at least one pocket. Erdős defined a *flip* as a rotation of a pocket's chain of edges into 3D about the pocket lid by 180°, landing the subchain back in the plane of the polygon, such that the polygon remains simple (i.e., non-self-intersecting); see Figure 6. He asked whether every polygon may be convexified by a finite number of simultaneous pocket flips.

The answer was provided in a later issue of the *Monthly* by de Sz.-Nagy [Nag39]. First, Nagy observed that flipping several pockets at once could lead to self-crossing; see Figure 7b. However, restricting to one flip at a time, Nagy proved that a finite number of flips suffice to convexify any polygon; see Figure 6 for a three-step example. This beautiful result has been rediscovered and reproved several times, as uncovered by Grünbaum and Toussaint, and detailed in their histories of the problem [Grü95, Tou99b], and only recently has a subtle oversight in Nagy's proof been corrected.

Unfortunately, the number of required flips can be arbitrarily large in terms of the number of vertices, even for a quadrangle. This fact was originally proved by Joss and Shannon (1973); see [Grü95, Tou99b, BDD$^+$01]. Figure 8 shows the construction. By making the vertical edge of the quadrangle very short and

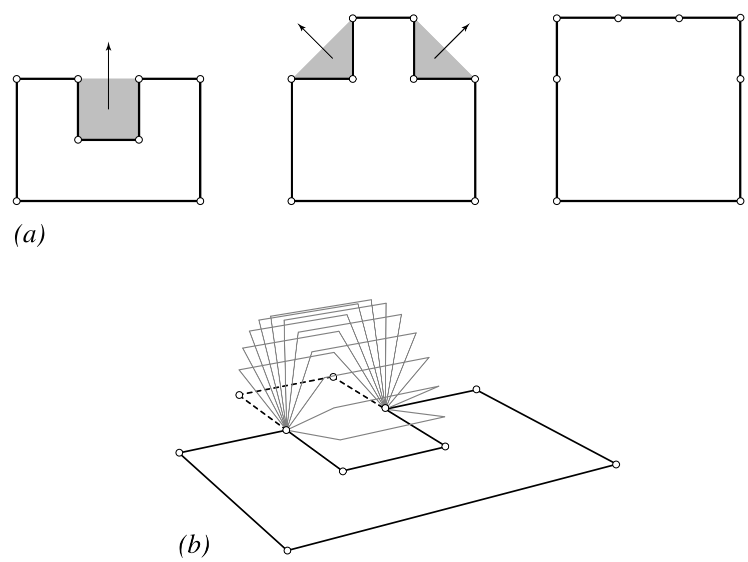


Figure 6: (a) Flipping a polygon until it is convex. Pockets are shaded. (b) The first flip shown in three dimensions.

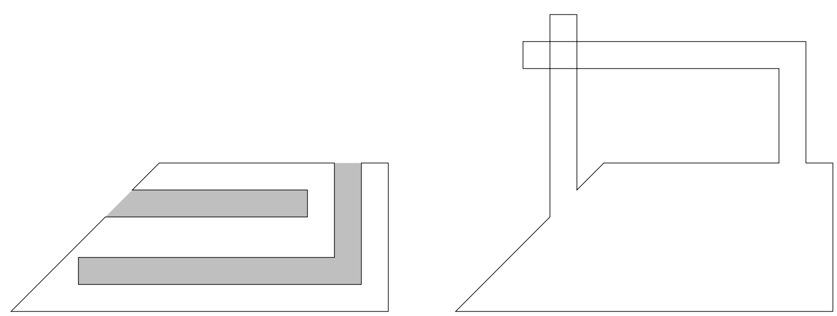


Figure 7: Flipping multiple pockets simultaneously can lead to crossings [Nag39].

even closer to the horizontal edge, the angles after the first flip approach the mirror image of the original quadrangle, and hence the number of required flips approaches infinity.

Mark Overmars[4] posed the still-open problem of bounding the number of flips in terms of natural measures of geometric closeness such as the diameter (maximum distance between two vertices), sharpest angle, or the minimum feature size (minimum distance between two nonincident edges).

Another open problem is to determine the complexity of finding the shortest or longest sequence of flips to convexify a given polygon. Weak NP-hardness

[4]Personal communication, February 1998.

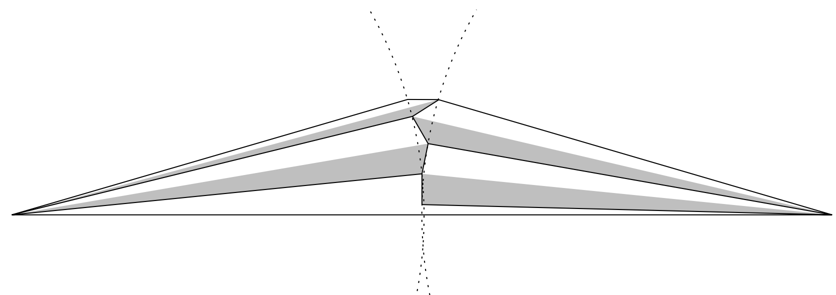


Figure 8: Quadrangles can require arbitrarily many flips to convexify [Grü95, Tou99b, BDD+01].

has been established for the related problem of finding the longest sequence of *flipturns* [ACD+02].

2.3.2 Flips in Nonsimple Polygons

Flips can be generalized to apply to nonsimple polygons: consider two vertices adjacent along the convex hull of the polygon, splitting the polygon into two chains, and rotate one (either) chain by 180° with respect to the other chain about the axis through the two vertices. Simplicity may not be preserved throughout the motion, just as it may not hold in the initial or final configuration. The obvious question is whether every nonsimple polygon can be convexified by a finite sequence of such flips. Grünbaum and Zaks [GZ98] proved that if at each step we choose the flip that maximizes the resulting sum of distances between all pairs of vertices, then this metric increases at each flip, and the polygon becomes convex after finitely many flips. Without sophisticated data structures, computing these flips requires $\Omega(n^2)$ time per flip. Toussaint [Tou99b] proved that a different sequence of flips convexifies a nonsimple polygon, and this sequence can be computed in $O(n)$ time per flip. More recently, it has been established[5] that every sequence of flips eventually convexifies a nonsimple polygon. We expect that each flip can be executed in polylogarithmic amortized time using dynamic convex-hull data structures as in [ACD+02].[6]

2.3.3 Deflations

A *deflation* [FHM+01, Weg93, Tou99b] is the reverse of a flip, in the sense that a deflation of a polygon should result in a simple polygon that can be flipped into the original polygon. More precisely, a deflation is a rotation by 180° about a line meeting the polygon at two vertices and nowhere else, thus separating the chain into two subchains, such that the rotation does not cause

[5] Personal communication with Therese Biedl, May 2001.
[6] Personal communication with Jeff Erickson.

any intersections. Hence, after the deflation, this line becomes a line of support (a line extending a convex-hull edge). Wegner [Weg93] proposed the notion of deflations, and their striking similarity to flips led him to conjecture that every polygon can be deflated only a finite number of times. Surprisingly, this is not true: Fevens, Hernandez, Mesa, Soss, and Toussaint [FHM+01] characterized a class of quadrangles whose unique deflation leads to another quadrangle in the class, thus repeating ad infinitum.

2.3.4 Other Variations

Erdős flips have inspired several directions of research on related notions, including pivots, pops, and flipturns. See [Tou99b] for a survey of this area, with more recent work on flipturns in [ABC+00, ACD+02, Bie05].

2.3.5 Efficient Algorithms for Planar Linkages in 3D

Motivated by the inefficiency of the flip algorithm, Biedl et al. [BDD+01] developed an algorithm to convexify planar polygons by motions in 3D using a linear number of simple moves. The essence of this algorithm is to lift the polygon, bar by bar, at all times maintaining a convex chain (or *arch*) lying in a plane orthogonal to the plane containing the polygon; see Figure 9. The details of the algorithm are significantly more involved than the overarching idea.

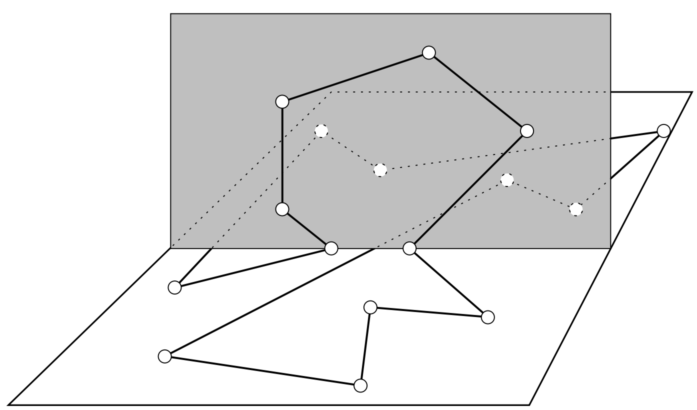


Figure 9: A planar polygon partially lifted into a convex arch lying in a vertical plane (shaded). (Based on Figure 6 of [BDD+01].)

A second linear-time algorithm, which is in some ways conceptually simpler, was developed by Aronov, Goodman, and Pollack [AGP02]. Their algorithm at all times maintains the arch as a convex quadrilateral. At each step, the algorithm lifts two edges, forming a “twisted trapezoid,” incorporates the trapezoid into the arch, makes the arch planar, and reduces it back to a quadrilateral. Avoiding intersections during the lifting phase requires a delicate argument.

In contrast to convexifying a cycle, it is relatively easy to straighten a polygonal arc lying in a plane, or on the surface of a convex polyhedron, by motions in 3D [BDD+01]. For an arc in a plane, the basic idea is to pull the arc up into a vertical line. For a convex surface, the same idea is followed, but with the orientation of the line changing to remain normal to the surface. The algorithm lifts each bar in turn, from one end of the arc to the other, at all times maintaining a prefix of the arc in a line normal to the current facet of the polyhedron. Each lifting motion causes two joint angles to rotate, so that the lifted prefix remains normal to the facet at all times, while the remainder of the chain remains in its original position. Whenever the algorithm reaches a vertex that bridges between two adjacent facets, it rotates the prefix to bring it normal to the next facet. This algorithm also generalizes to flattening planar trees and trees on the surface of a convex polyhedron, via motions in 3D.

2.3.6 Almost Knots

What if the linkage starts in an arbitrary position in 3D instead of in a plane? In general, a polygonal arc or an unknotted polygonal cycle in 3D cannot always be straightened or convexified [CJ98, Tou01, BDD+01] (Table 1, 3D row). Figure 10 shows an example of a locked arc in 3D. Provided that each of the two end bars is longer than the sum s of the middle three bar lengths, the ends of the chain cannot get close enough to the middle bars to untangle the chain (sometimes called the "knitting needles" example). More precisely, because the ends of the chain remain outside a sphere with radius s and centered at one of the middle vertices, we can connect the ends of the chain with an unknotted flexible cord outside the sphere, and any straightening motion unties the resulting knot, which is impossible without crossings [BDD+01].

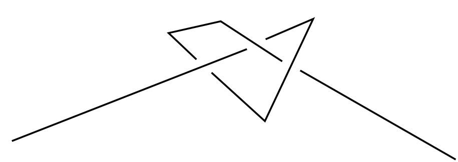


Figure 10: A locked polygonal arc in 3D with 5 bars [CJ98, BDD+01].

Alt, Knauer, Rote, and Whitesides [AKRW04] proved that it is PSPACE-hard to decide whether a 3D polygonal arc (or a 2D polygonal tree) can be reconfigured between two specified configurations. On the other hand, it remains open to determine the complexity of deciding whether a polygonal arc can be straightened. The next two sections describe special cases of 3D chains, more general than planar chains, that can be straightened and convexified.

2.3.7 Simple Projection

The "almost knottedness" of the example in Section 2.3.6 suggests that polygonal chains having simple orthogonal projections can always be straightened or convexified. This fact is established by two papers [BDD$^+$01, CKM$^+$01]. In addition, there is a polynomial-time algorithm to decide whether a polygonal chain has a simple projection, and if so find a suitable plane for projection [BGRT99].

For a polygonal arc with a simple orthogonal projection, the straightening method is relatively straightforward [BDD$^+$01]. The basic idea is to process the arc from one end to the other, accumulating bars into a compact "accordion" (x-monotone chain) lying in a plane orthogonal to the projection plane, in which each bar is nearly vertical. Once this accumulation is complete, the planar accordion is unfolded joint-by-joint into a straight arc. We observe that a similar algorithm can be used to fold a polygonal tree with a simple orthogonal projection into a generalized accordion, which can then be folded into a flat configuration.

For a polygonal cycle with a simple orthogonal projection, the convexification method is based on two steps [CKM$^+$01]. First, the projection of the polygon is convexified via the results described in Section 2.2.1, by folding the 3D polygon to track the shadow, keeping constant the ascent of each bar. Second, Calvo, Krizanc, Morin, Soss, and Toussaint [CKM$^+$01] develop an algorithm for convexifying a polygon with convex projection. The basic idea is to reconfigure the convex projection into a triangle, and stretch each accordion formed by an edge in the projection. In linear time they show how to compute a motion for the second step that consists of $O(n)$ simple moves, each changing at most seven vertex angles.

2.3.8 Interlocked Chains in 3D

Although Table 1 settles the question of when *one* chain can lock (only in 3D), the conditions that permit pairs of chains to "interlock" are largely unknown. This line of investigation was prompted by a question posed by Anna Lubiw [DO01]: into how many pieces must an n-bar 3D chain be cut (at vertices) so that the pieces can be separated and straightened? It is now known that the chain need be fractured into no more than $\lceil n/2 \rceil - 1$ pieces [DLOS02] but this upper bound is likely not tight: the only lower bound known is $\lfloor (n-1)/4 \rfloor$.

A collection of disjoint, noncrossing chains can be *separated* if, for any distance d, there is a non-self-crossing motion that results in every pair of points on different chains being separated by at least d. If a collection cannot be separated, its chains are *interlocked*. Which collections of relatively short chains can interlock was investigated in several papers [DLOS03, DLOS02]. Three typical results (all for chains with universal joints) are as follows:

1. No pair of 3-bar open chains can interlock, even with an arbitrary number of additional 2-bar open chains.
2. A 3-bar open chain can interlock with a 4-bar closed chain. (See Figure 11.)
3. A 3-bar open chain can interlock with a 4-bar open chain.

The proof of the first result (for just a pair of 3-bar chains) identifies a plane parallel to and separating the middle bars of each chain, and then nonuniformly scales the coordinate system to straighten the other links while avoiding intersections. The second result uses a topological argument based on "links" (multicomponent knots), in a manner similar to the use of knots in the proof that the chain in Figure 10 is locked. The proof of the third listed result is quite intricate, relying on ad hoc geometric arguments [DLOS02]. There are many open problems here, one of the most intriguing being this: what is the smallest k that permits a k-bar open chain to interlock with a 2-bar open chain [GLO$^+$04]?

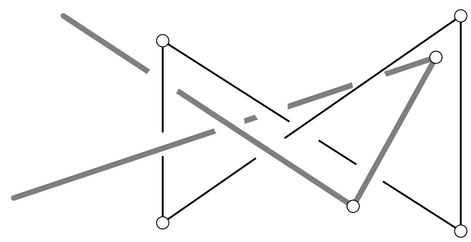


Figure 11: A 3-bar open chain (grey) interlocked with a 4-bar closed chain (black).

2.4 Fundamental Questions in 4D and Higher Dimensions

In all dimensions higher than 3, it is known that all knots are trivial; analogously, all polygonal arcs can be straightened, all polygonal cycles can be convexified, and all polygonal trees can be flattened [CO01] (Table 1, 4D row). Intuitively, this result holds because the number of degrees of freedom of any vertex is at least two higher than the dimensionality of the obstacles imposed by any bar. This property allows Cocan and O'Rourke [CO01] to establish, for example, that the last bar of a polygonal arc can be unfolded by itself to any target position that is simple.

Cocan and O'Rourke [CO01] show how to straighten an arc using $O(n)$ simple moves that can be computed in $O(n^2)$ time and $O(n)$ space. On the other hand, their method for convexifying a polygon requires $O(n^6)$ simple moves and $O(n^6 \log n)$ time to compute.

2.5 Protein Folding

Protein folding [CD93, Hay98, ML94] is an important problem in molecular biology because it is generally believed that the folded structure of a protein (the fundamental building block of life) determines its function and behavior.

2.5.1 Connection to Linkages

A protein can be modeled by a linkage in which the vertices represent amino acids and the bars represent bonds connecting them. The bars representing bonds are typically close in length, within a factor less than two. Depending on the level of detail, the protein can be modeled as a tree (more precise) or as a chain (less precise).

An amazing property of proteins is that they fold quickly and consistently to a minimum-energy configuration. Understanding this motion has immediate connections to linkage folding in 3D. A central unsolved theoretical question [BDD$^+$01] arising in this context is whether every equilateral polygonal arc in 3D can be straightened. Cantarella and Johnston [CJ98] proved that this is true for arcs of at most 5 bars. More generally, can every equilateral polygonal tree in 3D be flattened?

2.5.2 Fixed-Angle Linkages

A more accurate mathematical model of foldings of proteins is not by linkages whose vertices are universal joints, but rather by *fixed-angle* linkages in which each vertex forms a fixed angle between its incident bars. This angular constraint roughly halves the number of degrees of freedom in the linkage; the basic motion is rotating a portion of the linkage around a bar of the linkage. Foldings of such linkages have been explored extensively by Soss and Toussaint [ST00, Sos01]. For example, they prove in [ST00] that it is NP-complete to decide whether a fixed-angle polygonal arc can be flattened (reconfigured to lie a plane), and in [Sos01] that it is NP-complete to decide whether a fixed-angle polygonal arc can be folded into its mirror image. More positive results analyze the polynomial complexity of determining the maximum extent of a rotation around a bar: Soss and Toussaint [ST00, Sos01] prove an $O(n^2)$ upper bound, and Soss, Erickson, and Overmars [Sos01, SEO03] give a 3SUM-hardness reduction, suggesting an $\Omega(n^2)$ lower bound.

Another line of investigation on fixed-angle chains was opened in [ADD$^+$02, ADM$^+$02]. Define a linkage X to be *flat-state connected* if, for each pair of its flat realizations x_1 and x_2, there is a reconfiguration from x_1 to x_2 that avoids self-intersection throughout. In general this motion alters the linkage to nonflat configurations in $\mathbb{R}^3$ intermediate between the two flat states. The main question is to determine whether every fixed-angle open chain is flat-state connected. It has been established that the answer is YES for chains all of whose fixed angles between consecutive bars are nonacute [ADD$^+$02], and although other special cases have been settled [ADM$^+$02], the main question remains open.

2.5.3 Producible Chains

A connection between fixed-angle nonacute chains and a model of protein production was recently established in [DLO03]. Here the ribosome—the "machine" that creates protein chains in biological cells—is modeled as a cone, with the fixed-angled chain produced bar-by-bar inside and emerging through the cone's apex. A configuration of a chain is said to be *α-producible* if there exists a continuous motion of the chain as it is created by the above model from within a cone of half-angle $\alpha \leq \pi/2$. The main result of [DLO03] is a theorem that identifies producible with flattenable chains, in this sense: a configuration of a chain whose fixed angles are $\geq \pi - \alpha$, for $\alpha \leq \pi/2$, is α-producible if and only if it is flattenable. For example, for $\alpha = 45°$, this theorem says that a fixed-135°-angle chain (which is nonacute) is producible within a 90° cone if and only if that configuration is flattenable.

The proof uses a coiled cannonical configuration of the chain, which can be obtained by time-reversal of the production steps, winding the chain inside the cone. This canonical form establishes that all α-producible chains can be reconfigured to one another. Then it is shown how to produce any flat configuration by rolling the cone around on the plane into which the flat chain is produced. Because locked chains are not flattenable, the equivalence of producible and flattenable configurations shows that cone production cannot lead to locked configurations. This result in turn leads to the conclusion that the producible chains are rare, in a technical sense, suggesting that the entire configuration space for folding proteins might not need to be searched.

2.5.4 H-P Model

So far in this section we have not considered the forces involved in protein folding in nature. There are several models of these forces.

One of the most popular models of protein folding is the hydrophobic-hydrophilic (H-P) model [CD93, Dil90, Hay98], which defines both a geometry and a quality metric of foldings. This model represents a protein as a chain of amino acids, distinguished into two categories, hydrophobic (H) and hydrophilic (P). A folding of such a protein chain in this model is an embedding along edges of the square lattice in 2D or the cubic lattice in 3D without self-intersection. The optimum or minimum-energy folding maximizes the number of hydrophobic (H) nodes that are adjacent in the lattice. Intuitively, this metric causes hydrophobic amino acids to avoid the surrounding water.

This combinatorial model is attractive in its simplicity, and already seems to capture several essential features of protein folding such as the tendency for the hydrophobic components to fold to the center of a globular protein [CD93]. While a 3D H-P model most naturally matches the physical world, in fact it is more realistic as a 2D model for computationally feasible problem sizes. The reason for this is that the perimeter-to-area ratio of a short 2D chain is a close approximation to the surface-to-volume ratio of a long 3D chain [CD93, Hay98].

Much work has been done on the H-P model [BL98, CD91, CD90, CGP$^+$98, HI96, LD89, LD90, LW91, UM93a, UM93b, UM93c]. Recently, on the computational side, Berger and Leighton [BL98] proved NP-completeness of finding the optimal folding in 3D, and Crescenzi et al. [CGP$^+$98] proved NP-completeness in 2D. Hart and Istrail [HI96] have developed a 3/8-approximation in 3D and a 1/4-approximation in 2D for maximizing the number of hydrophobic-hydrophobic adjacencies.

Aichholzer, Bremner, Demaine, Meijer, Sacristan, and Soss [ABD$^+$03] have begun exploring an important yet potentially more tractable aspect of protein folding: can we design a protein that folds stably into a desired shape? In the H-

P model, a protein folds *stably* if it has a unique minimum-energy configuration. So far, Aichholzer et al. [ABD$^+$03] have proved the existence of stably folding proteins of all lengths divisible by 4, and for closed chains of all possible (even) lengths. It remains open to characterize the possible shapes (connected subsets of the square grid) attained by stable protein foldings.

3 Paper

Paper folding (origami) has led to several interesting mathematical and computational questions over the past fifteen years or so. A piece of paper, normally a (solid) polygon such as a square or rectangle, can be folded by a continuous motion that preserves the distances on the surface and does not cause the paper to properly self-intersect. Informally, paper cannot tear, stretch, or cross itself, but may otherwise bend freely. (There is a contrast here to folding other materials, such as sheet metal, that must remain piecewise planar throughout the folding process.) Formally, a folding is a continuum of isometric embeddings of the piece of paper in $\mathbb{R}^3$. However, the use of the term "embedding" is weak: paper is permitted to touch itself provided it does not properly cross itself. In particular, a *flat folding* folds the piece of paper back into the plane, and so the paper must necessarily touch itself. We frequently ignore the continuous motion of a folding and instead concentrate on the final folded state of the paper; in the case of a flat folding, the flat folded state is called a *flat origami*. This concentration on the final folded state was recently justified by a proof that there always exists a continuous motion from a planar polygonal piece of paper to any "legal" folded state [DDMO04].

Some of the pioneering work in origami mathematics (see Section 3.3.1) studies the *crease pattern* that results from unfolding a flat origami, that is, the graph of edges on the paper that fold to edges of a flat origami. Stated in reverse, what crease patterns have flat foldings? Various necessary conditions are known [Hul94, Jus94, Kaw89], but there is little hope for a polynomial characterization: Bern and Hayes [BH96] have shown that this decision problem is NP-hard.

A more recent trend, as in [BH96], is to explore *computational origami*, the algorithmic aspects of paper folding. This field essentially began with Robert Lang's work on algorithmic origami design [Lan96], starting around 1993. Since then, the field of computational origami has grown significantly, in particular in the past two years by applying computational geometry techniques. This section surveys this work. See also [DD01].

3.1 Categorization

Most results in computational origami fall under one or more of three categories: universality results, efficient decision algorithms, and computational intractability results. This categorization applies more generally to folding and unfolding, but is particularly useful for results in computational origami.

A *universality result* shows that, subject to a certain model of folding, everything is possible. For example, any tree-shaped origami base (Section 3.2.2), any polygonal silhouette (Section 3.2.1), and any polyhedral surface (Section 3.2.1) can be folded out of a sufficiently large piece of paper. Universality results often come with efficient algorithms for finding the foldings; pure existence results are rare.

When universality results are impossible (some objects cannot be folded), the next-best result is an *efficient decision algorithm* to determine whether a given object is foldable. Here "efficient" normally means "polynomial time." For example, there is a polynomial-time algorithm to decide whether a "map" (grid of creases marked mountain and valley) can be folded by a sequence of "simple folds" (Section 3.3.4).

Not all paper-folding problems have efficient algorithms, and this can be proved by a *computational intractability result*. For example, it is NP-hard to tell whether a given crease pattern folds into some flat origami (Section 3.3.2), even when folds are restricted to simple folds (Section 3.3.4). These results imply that there are no polynomial-time algorithms for these problems, unless some of the hardest computational problems known can also be solved in polynomial time, which is generally deemed unlikely.

We further distinguish computational origami results as addressing either *origami design* or *origami foldability*. In origami design, some aspects of the target configuration are specified, and the goal is to design a suitable detailed folded state that can be folded out of paper. In origami foldability, the target configuration is unspecified and arbitrary; rather, the initial configuration is specified, in particular the crease pattern, possibly marked with mountains and valleys, and the goal is to fold something (anything) using precisely those creases. While at first it may seem that understanding origami foldability is a necessary component for origami design, the results indicate that in fact origami design is easier to solve than origami foldability, which is usually intractable.

Our survey of computational origami is partitioned accordingly into Section 3.2 (origami design) and Section 3.3 (origami foldability).

3.2 Origami Design

We define *origami design* loosely as, given a piece of paper, fold it into an object with certain desired properties, e.g., a particular shape. The natural theoretical version of this problem is to ask for an origami with a specific silhouette or three-dimensional shape; this problem can be solved in general (Section 3.2.1), although the algorithms developed so far do not lead to practical foldings. A specific form of this problem has been solved for practical purposes by Lang's tree method (Section 3.2.2), which has brought modern origami design to a new level of complexity. Related to this work is the problem of folding a piece of paper to align a prescribed graph (Section 3.2.3), which can be used for a magic trick involving folding and one complete straight cut.

3.2.1 Silhouettes and Polyhedra

A direct approach to origami design is to specify the exact final shape that the paper should take. More precisely, suppose we specify a particular flat silhouette, or a three-dimensional polyhedral surface, and desire a folding of a sufficiently large square of paper into precisely this object, allowing coverage by multiple layers of paper. For what polyhedral shapes is this possible? This problem is implicit throughout origami design, and was first formally posed by Bern and Hayes in 1996 [BH96]. The surprising answer is “always,” as established by Demaine, Demaine, and Mitchell in 1999 [DDM99, DDM00].

The basic idea of the approach is to fold the piece of paper into a thin strip, and then wrap this strip around the desired shape. This wrapping can be done particularly efficiently using methods in computational geometry. Specifically, three algorithms are described in [DDM00] for this process. One algorithm optimizes paper usage: the amount of paper required can be made arbitrarily close to the surface area of the shape, but only at the expense of increasing the aspect ratio of the rectangular paper. Another algorithm maximizes the width of the strip subject to some constraints. A third algorithm places the visible *seams* of the paper in any desired pattern forming a decomposition of the sides into convex polygons. In particular, the number and total length of seams can be optimized in polynomial time in most cases [DDM00].

All of these algorithms allow an additional twist: the paper may be colored differently on both sides, and the shape may be two-colored according to which side should be showing. In principle, this allows the design of two-color models similar to the models in Montroll's *Origami Inside-Out* [Mon93]. An example is shown in Figure 12.

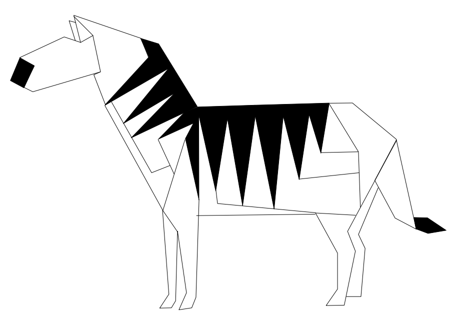


Figure 12: A flat folding of a square of paper, black on one side and white on the other side, designed by John Montroll [Mon91, pp. 94–103]. (Figure 1(b) of [DDM00].)

Because of the use of thin strips, none of these methods lead to practical foldings, except for small examples or when the initial piece of paper is a thin strip. Nonetheless, the universality results of [DDM00] open the door to many new problems. For example, how small a square can be folded into a desired object, e.g., a $k \times k$ chessboard? This optimization problem remains open even in this special case, as do many other problems about finding efficient, practical foldings of silhouettes, two-color patterns, and polyhedra.

3.2.2 Tree Method

The *tree method of origami design* is a general approach for "true" origami design (in contrast to the other topics that we discuss, which involve less usual forms of origami). In short, the tree method enables design of efficient and practical origami within a particular class of three-dimensional shapes, most useful for origami design. Some components of this method, such as special cases of the constituent molecules and the idea of disk packing, as well as other methods for origami design, have been explored in the Japanese technical origami community, in particular by Jun Maekawa, Fumiaki Kawahata, and Toshiyuki Meguro. This work has led to several successful designs, but a full survey is beyond the scope of this paper; see [Lan03, Lan98]. It suffices to say that the explosion in origami design over the last 30 years, during which the majority of origami models have been designed, may largely be due to an understanding of these general techniques.

Here we concentrate on Robert Lang's work [Lan94a, Lan94b, Lan96, Lan98, Lan03], which is the most extensive. Over the past decade, starting around 1993, Lang has developed the tree method to the point where an algorithm and computer program have been explicitly defined and implemented. Anyone with a Macintosh computer can experiment with the tree method using Lang's program TreeMaker [Lan98].

The tree method allows one to design an origami *base* in the shape of a specified tree with desired edge lengths, which can then be folded and shaped into an origami model. See Figure 13 for an example. More precisely, the tree method designs a *uniaxial base* [Lan96], which must have the following properties: the base lies above and on the xy-plane, all facets of the base are perpendicular to the xy-plane, the projection of the base to the xy plane is precisely where the base comes in contact with the xy-plane, and this projection is a one-dimensional tree.

It is known that every metric tree (unrooted tree with prescribed edge lengths) is the projection of a uniaxial base that can be folded from, e.g., a square. The tree method gives an algorithm to find the folding that is optimal in the sense that it folds the uniaxial base with the specified projection using the smallest possible square piece of paper (or more generally, using the smallest possible scaling of a given convex polygon). These foldings have led to many impressive origami designs; see [Lan03] in particular.

There are two catches to this result. First, it is currently unknown whether the prescribed folding self-intersects, though it is conjectured that self-intersection does not arise, and this conjecture has been verified on extensive examples. Second, the optimization problem is difficult, a fairly general form of nonlinear constrained optimization. So while optimization is possible in principle in finite time, in practice heuristics must be applied; fortunately, such heuristics frequently yield good, practical solutions. Indeed, additional practical constraints

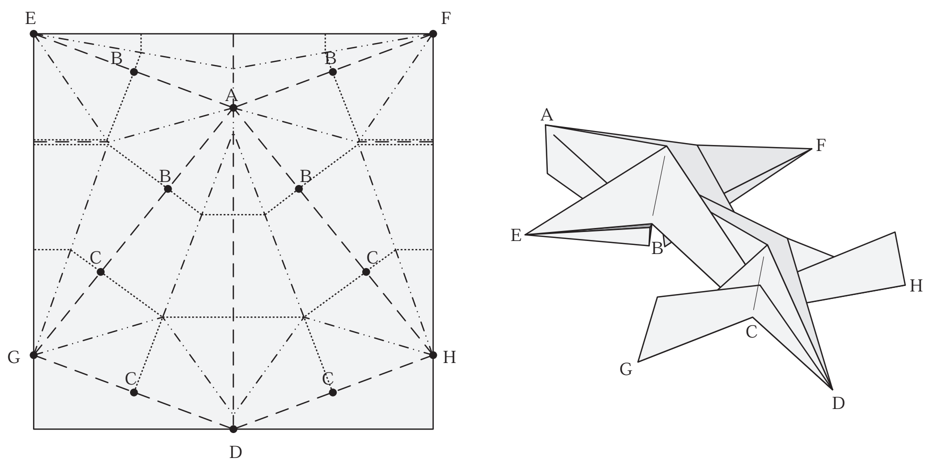


Figure 13: Lang's TreeMaker applied to an 8-vertex tree to produce a lizard base. (Figure 2.1.11 of [Lan98].)

can be imposed, such as symmetry in the crease pattern, or the constraint that angles of creases are integer multiples of some value (e.g., 22.5°) subject to some flexibility in the metric tree.

3.2.3 One Complete Straight Cut

Take a piece of paper, fold it flat, make one complete straight cut, and unfold the pieces. What shapes can result? This *fold-and-cut* problem was first formally stated by Martin Gardner in 1960 [Gar95]. The problem has a much longer history, going back to 1721 in a Japanese puzzle book [Sen21], Betsy Ross in 1777 [Har73], and Houdini in 1922 [Hou22]. See [DDL98] for a more detailed history.

More formally, given a planar graph drawn with straight edges on a piece of paper, can the paper be folded flat so as to map the entire graph to a common line, and map nothing else to that line? The surprising answer is that this is always possible, for any collection of line segments in the plane, forming nonconvex polygons, adjoining polygons, nested polygons, etc. There are two solutions to the problem. The first (partial) solution [DDL98, DDL99b] is based on a structure called the straight skeleton, which captures the symmetries of the graph, thereby exploiting a more global structure of the problem. This solution applies to a large class of instances, which we do not describe in detail here. See Figure 14 for two examples. The second (complete) solution [BDEH01] is based on disk packing to make the problem more local, and achieves efficient bounds on the number of creases.

While this problem may not seem directly connected to pure paper folding because of the one cut, the equivalent problem of folding a piece of paper to line up a given collection of edges is in fact closely connected to origami design.

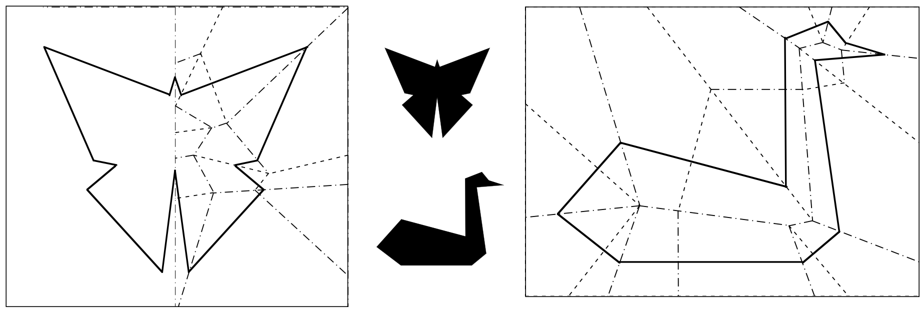


Figure 14: Crease patterns for folding a rectangle of paper flat so that one complete straight cut makes a butterfly (left) or a swan (right), based on [DDL98, DDL99b].

Specifically, one subproblem that arises in TreeMaker (Section 3.2.2) is that the piece of paper is decomposed into convex polygons, and the paper must be folded flat so as to line up all the edges of the convex polygons, and place the interior of these polygons above this line. The fold-and-cut problem is a generalization of this situation to arbitrary graphs: nonconvex polygons, nested polygons, etc. In TreeMaker, there are important additional constraints in how the edges can be lined up, called path constraints, which are necessary to enforce the desired geometric tree. These constraints lead to additional components in the solution called *gussets*.

3.3 Origami Foldability

We distinguish origami design from *origami foldability* in which the starting point is a given crease pattern and the goal is to fold an origami that uses precisely these creases. (Arguably, this is a special case of our generic definition of origami design, but we find it a useful distinction.) The most common case studied is when the resulting origami should be flat, i.e., lie in a plane.

3.3.1 Local Foldability

For crease patterns with a single vertex, it is relatively easy to characterize flat foldability. Without specified crease directions, a single-vertex crease pattern is flat-foldable precisely if the alternate angles around the vertex sum to 180°; see Figure 15. This is known as Kawasaki's theorem [BH96, Hul94, Jus94, Kaw89]. When the angle condition is satisfied, a characterization of valid mountain-valley assignments and flat foldings can be found in linear time [BH96, Jus94], using Maekawa's theorem [BH96, Hul94, Jus94] and another theorem of Kawasaki [BH96, Hul94, Kaw89] about constraints on mountains and valleys. In particular, Hull has shown that the number of distinct mountain-valley assignments of a vertex can be computed in linear time [Hul03].

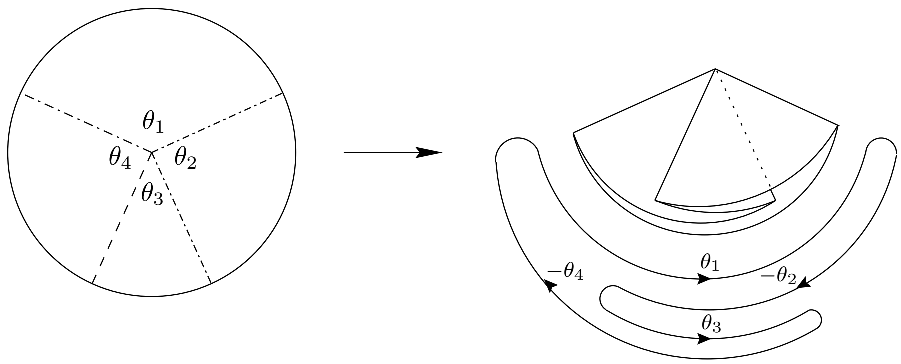


Figure 15: A locally flat-foldable vertex: $\theta_1 + \theta_3 + \cdots = \theta_2 + \theta_4 + \cdots = 180°$.

A crease pattern is called *locally foldable* if there is a mountain-valley assignment so that each vertex locally folds flat, i.e., a small disk around each vertex folds flat. Testing local foldability is nontrivial because each vertex has flexibility in its assignment, and these assignments must be chosen consistently: no crease should be assigned both mountain and valley by the two incident vertices. Bern and Hayes [BH96] proved that consistency can be resolved efficiently when it is possible: local foldability can be tested in linear time.

3.3.2 Existence of Folded States

Given a crease pattern, does it have a flat folded state? Bern and Hayes [BH96] have proved that this decision problem is NP-hard, and thus computationally intractable. Because local foldability is easy to test, the only difficult part is global foldability, or more precisely, computing a valid *overlap order* of the crease faces that fold to a common portion of the plane. Indeed, Bern and Hayes [BH96] prove that, given a crease pattern and a mountain-valley assignment that definitely folds flat, finding the overlap order of a flat folded state is NP-hard.

3.3.3 Equivalence to Continuous Folding Process

In the previous section we have alluded to the difference between two models of folding: the final folded state (specified by a crease pattern, mountain-valley or angle assignment, and overlap order) and a continuous motion to bring the paper to that folded state. Basically all results, in particular those described so far, have focused on the former model: proving that a folded state exists with the desired properties. Intuitively, by appropriately flexing the paper, any folded state can be reached by a continuous motion, so the two models should be equivalent. Only recently has this been proved, initially for rectangular pieces of paper [DM01], and recently for general polygonal pieces of paper [DDMO04] During these motions, the number of creases is finite at any moment in time, but overall the number of creases is uncountably infinite. An interesting open problem is whether a finite crease pattern suffices.

The only other paper of which we are aware that explicitly constructs continuous folding processes is [DD97]. This paper proves that every convex polygon can be folded into a uniaxial base via Lang's universal molecule [Lan98] without gussets. Furthermore, unlike [DM01], no additional creases are introduced during the motion, and each crease face remains flat. This result can be used to animate the folding process.

3.3.4 Map Folding: Sequence of Simple Folds

In contrast to the complex origami folds arising from reaching folded states [DD97, DM01], we can consider the less complex model of simple folds. A *simple fold* (or *book fold*) is a fold by $\pm 180°$ along a single line. Examples are shown in Figure 16. This model is closely related to "pureland origami" introduced by Smith [Smi76, Smi93].

We can ask the same foldability questions for a sequence of simple folds. Given a crease pattern, can it be folded flat via a sequence of simple folds? What if a particular mountain-valley assignment is imposed?

An interesting special case of these problems is *map folding* (see Figure 16): given a rectangle of paper with horizontal and vertical creases, each marked mountain or valley, can it be folded flat via a sequence of simple folds? Traditionally, map folding has been studied from a combinatorial point of view; see, e.g., [Lun68, Lun71]. Arkin, Bender, Demaine, Demaine, Mitchell, Sethia, and Skiena [ABD+04] have shown that deciding foldability of a map by simple folds can be solved in polynomial time. If the simple folds are required to fold all layers at once, the running time is at most $O(n \log n)$, and otherwise the running time is linear.

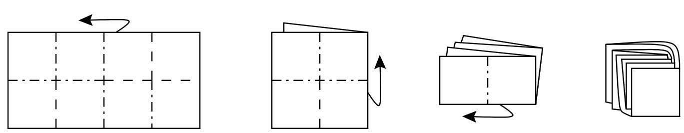


Figure 16: Folding a 2×4 map via a sequence of 3 simple folds.

Surprisingly, slight generalizations of map folding are (weakly) NP-complete [ABD+04]. Deciding whether a rectangle with horizontal, vertical, and diagonal ($\pm 45°$) creases can be folded via a sequence of simple folds is NP-complete. Alternatively, if the piece of paper is more general, a polygon with horizontal and vertical sides, and the creases are only horizontal and vertical, the same problem is NP-complete.

These hardness results are *weak* in the sense that they leave open the existence of a *pseudopolynomial-time* algorithm, whose running time is polynomial in the total length of creases. Another intriguing open problem, posed by Jack Edmonds, is the complexity of deciding whether a map has some flat folded state, as opposed to a folding by a sequence of simple folds. Examples of maps in which these two notions of foldability differ are shown in Figure 17.

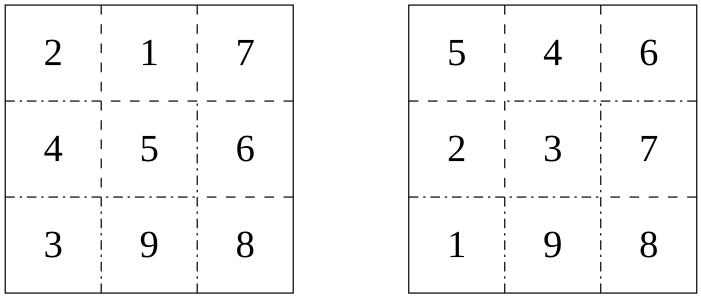


Figure 17: Two maps that cannot be folded by simple folds, but can be folded flat. (These are challenging puzzles.) The numbering indicates the overlap order of faces. (Figure 12 of [ABD$^+$04].)

3.4 Flattening Polyhedra

When one flattens a cardboard box for recycling, generally the surface is cut open. Suppose instead of allowing cuts to a polyhedral surface in order to flatten it, we treat it as a piece of paper and fold as in origami. We run into the same dichotomy as in Section 3.3.2: do we want a continuous motion of the polyhedron, or does a description of the final folded state suffice? If we start with a convex polyhedron, and each face of the crease pattern must remain rigid during the folding, then Connelly's extension [Con80] of Cauchy's rigidity theorem from 1813 [Cau13, Cro97] says that the polyhedron cannot fold at all. Even if we start with a nonconvex polyhedron and keep each face of the crease pattern rigid, the Bellows Theorem [CSW97] says that the volume of the polyhedron cannot change, so foldings are limited. However, if we allow the paper to curve (e.g., introduce new creases) during the motion, as in origami, then folding becomes surprisingly flexible. For example, a cone can be inverted [Con93]; see Figure 18.

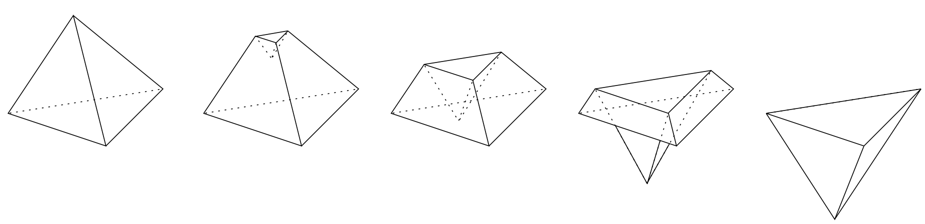


Figure 18: Inverting a tetrahedral cone by a continuous isometric motion. Based on Figure 2.5 of [Con93].

A natural question is whether every polyhedron can be *flattened*: folded into a flat origami. Intuitively, this can be achieved by applying force to the polyhedral model, but in practice this can easily lead to tearing. There is an interesting connection of this problem to a higher-dimensional version of the fold-and-cut problem from Section 3.2.3. Given any polyhedral complex, can $\mathbb{R}^3$ be folded (through $\mathbb{R}^4$) "flat" into $\mathbb{R}^3$ so that the surface of the polyhedral complex maps to a common plane, and nothing else maps to that plane? While the applicability of four dimensions is difficult to imagine, the problem's restriction to the surface of the complex is quite practical, e.g., in packaging: *flatten* the polyhedral complex into a flat folded state, without cutting or stretching the paper.

The flattening problem remains open if we desire a continuous folding process into the flat state. If we instead focus on the existence of a flat folded state of a polyhedron, then much more is known. Demaine, Demaine, and Lubiw[7] have shown how to flatten several classes of polyhedra, including convex polyhedra and orthogonal polyhedra. See Figure 19 for an example. Recently, Demaine, Demaine, Hayes, and Lubiw[8] have shown that all polyhedra have flat folded states. They conjecture further that every polyhedral complex can be flattened.

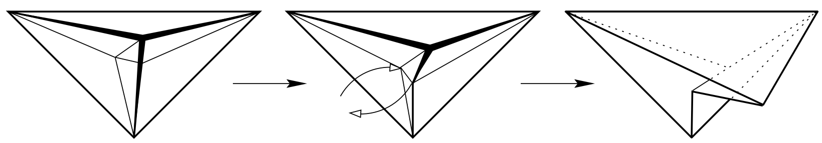


Figure 19: Flattening a tetrahedron, from left to right. Note that the faces are not flat in the middle picture.

A natural question is whether the methods of Demaine and Mitchell [DM01] and [DDMO04] described in Section 3.3.3 can be generalized to show that these folded states induce continuous folding motions as in Figure 18.

4 Polyhedra

A standard method for building a model of a polyhedron is to cut out a flat *net* or *unfolding*, fold it up, and glue the edges together so as to make precisely the desired surface. Given the polyhedron of interest, a natural problem is to find a suitable unfolding. On the other hand, given a polygonal piece of paper, we might ask whether it can be folded and its edges can be glued together so as to form a convex polyhedron. These two questions are addressed in Sections 4.1 and 4.2, respectively. Section 4.3 extends different forms of the latter question to nonconvex polyhedra. Section 4.4 connects these problems to linkage and paper folding.

4.1 Unfolding Polyhedra

A classic open problem is whether (the surface of) every convex polyhedron can be cut along some of its edges and unfolded into one flat piece without

[7] Manuscript, March 2001.
[8] Manuscript in preparation.

overlap [She75, O'R98]. Such *edge-unfoldings* go back to Dürer [Dür77], and have important practical applications in manufacturing, such as sheet-metal bending [O'R98, Wan97]. It seems folklore that the answer to this question should be YES, but the evidence for a positive answer is actually slim. Only very simple classes of polyhedra are known to be edge-unfoldable; for example, pyramids, prisms, "prismoids,"[9] and other more specialized classes [DO]. In contrast, experiments by Schevon [Sch89, O'R98] suggest that a random edge-unfolding of a random polytope overlaps with probability 1. Of course, such a result would not preclude, for every polytope, the existence of at least one nonoverlapping edge-unfolding, or even that a large but subconstant fraction of the polytope's edge-unfoldings do not overlap. However, the unlikeliness of finding an unfolding by chance makes the search more difficult.

An easier version of this edge-unfolding problem is the *fewest-nets* problem: prove an upper bound on the number of pieces required by a multipiece non-overlapping edge unfolding of a convex polyhedron. The obvious upper bound is the number F of faces in the polyhedron; the original problem asks whether an upper bound of 1 is possible. The first bound of cF for $c < 1$ was obtained by Michael Spriggs,[10] who established $c = 2/3$. The smallest value of c obtained so far[11] is $1/2$. Proving an upper bound that is sublinear in F would be a significant advancement.

We can also examine to what extent edge unfoldings can be generalized to nonconvex polyhedra. In particular, define a polyhedron to be *topologically convex* if its 1-skeleton (graph) is the 1-skeleton of a convex polyhedron. Does every topologically convex polyhedron have an edge-unfolding? In particular, every polyhedron composed of convex faces and homeomorphic to a sphere is topologically convex; can they all be edge-unfolded? This problem was posed by Schevon [Sch87].

Bern, Demaine, Eppstein, Kuo, Mantler, and Snoeyink [BDE$^+$03] have shown that the answer to both of these questions is NO: there is a polyhedron composed of triangles and homeomorphic to a sphere that has no (one-piece, nonoverlapping) edge-unfolding. The polyhedron is shown in Figure 20. It consists of four "hats" glued to the faces of a regular tetrahedron, such that only the peaks of the hats have positive curvature, that is, have less than 360° of incident material. This property limits the unfoldings significantly, because (1) any set of cuts must avoid cycles in order to create a one-piece unfolding, and (2) a leaf in a forest of cuts can only lie at a positive-curvature vertex of the polyhedron: a leaf at a negative-curvature vertex (more than 360° of incident material) would cause local overlap.

The complexity of deciding whether a given topologically convex polyhedron can be edge-unfolded remains open.

Another intriguing open problem in this area is whether every polyhedron

[9] The convex hull of two equiangular convex polygons, oriented so that corresponding edges are parallel.

[10] Personal communication, August 2003.

[11] Personal communication from Vida Dujmović, Pat Morin, and David Wood, February 2004.

homeomorphic to a sphere has *some* one-piece unfolding, not necessarily using cuts along edges. It is known that every convex polyhedron has an unfolding in this model, allowing cuts across the faces of the polytope. Specifically, the *star unfolding* [AAOS97, AO92] cuts the set of shortest paths from a common source point to each vertex of the polytope, and the *source unfolding* [MMP87] cuts the set of points with more than one shortest path to a common source. Both of these unfoldings avoid overlap, the star unfolding being the more difficult case to establish [AO92]. The source unfolding (but not the star unfolding) also generalizes to unfold convex polyhedra in higher dimensions [MP03a].

But many nonconvex polyhedra also have such unfoldings. For example, Figure 20 illustrates one for the polyhedron described above. Biedl, Demaine, Demaine, Lubiw, Overmars, O'Rourke, Robbins, and Whitesides [BDD+98] have shown how to unfold many orthogonal polyhedra, even with holes and knotted topology, although it remains open whether all orthogonal polyhedra can be unfolded. The only known scenario that prevents unfolding altogether [BDE+03] is a polyhedron with a single vertex of negative curvature (see Figure 21), but this requires the polyhedron to have boundary (edges incident to only one face).

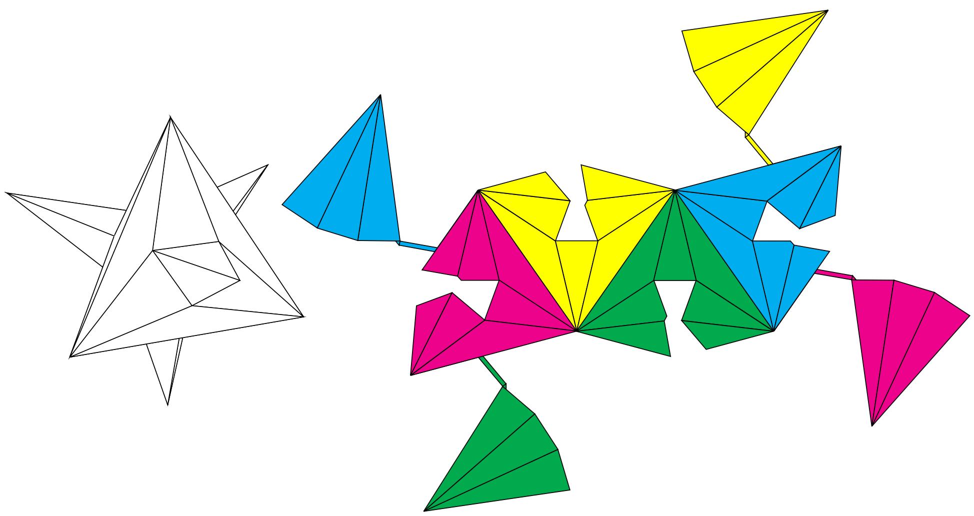


Figure 20: (Left) Simplicial polyhedron with no edge-unfolding. (Right) An unfolding when cuts are allowed across faces.

A recent approach to unfolding both convex and nonconvex polyhedra in any dimension is the notion of "vertex-unfolding" [DEE+01]; see Figure 22. Specifically, a *vertex-unfolding* may cut only along edges of the polyhedron (like an edge-unfolding) but permits the facets to remain connected only at vertices (instead of along edges as in edge-unfolding). Thus, a vertex-unfolding is connected, but its interior may be disconnected, "pinching" at a vertex. This notion also generalizes to polyhedra in any dimension. Demaine, Eppstein, Erickson, Hart, and O'Rourke [DEE+01] proved that every simplicial manifold in any di-

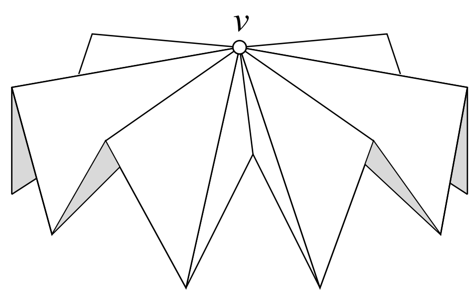


Figure 21: A polyhedron with boundary that has no one-piece unfolding even when cuts are allowed across faces. Vertex v has negative curvature, that is, more than 360° of incident material. (Based on Figure 9 of [BDE$^+$03].)

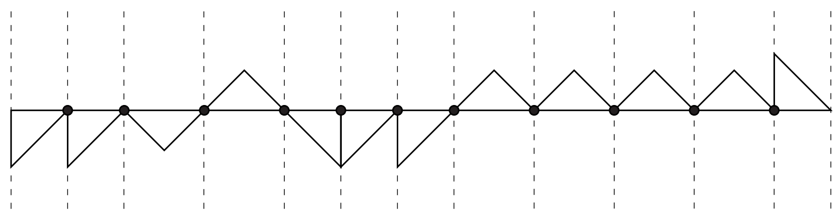


Figure 22: Vertex-unfolding of a triangulated cube with hinge points aligned. (Based on Figure 2 of [DEE$^+$01].)

mension has a nonoverlapping vertex-unfolding. In particular, this result covers triangulated polyhedra in 3D, possibly with boundary, but it remains open to what extent vertex-unfoldings exist for polyhedra with nontriangular faces. For example, does every convex polyhedron in 3D have a vertex-unfolding?

4.2 Folding Polygons into Convex Polyhedra

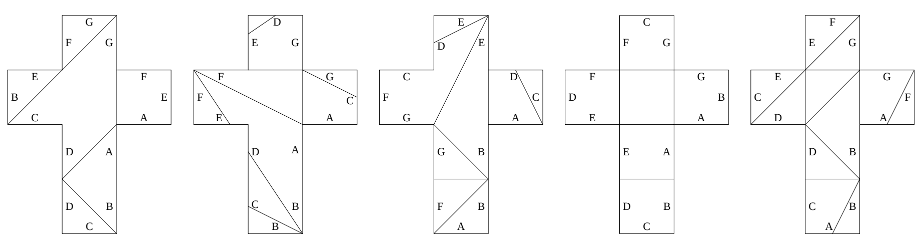
In addition to unfolding polyhedra into simple planar polygons, we can consider the reverse problem of folding polygons into polyhedra. More precisely, when can a polygon have its boundary glued together, with each portion gluing to portions of matching length, and the resulting topological object be folded into a *convex* polyhedron? (There is almost too much flexibility with nonconvex polyhedra for this problem, but see Section 4.3 for related problems of interest in this context.) A particular kind of gluing is an *edge-to-edge* gluing, in which each entire edge of the polygon is glued to precisely one other edge of the polygon. The existence of such a gluing requires a perfect pairing of edges with matching lengths.

4.2.1 Edge-to-Edge Gluings

Introducing this area, Lubiw and O'Rourke [LO96] showed how to test in polynomial time whether a polygon has an edge-to-edge gluing that can be folded into a convex polyhedron, and how to list all such edge-to-edge gluings in exponential time. A key tool in their work is a theorem of A. D. Aleksandrov [Ale50]. The theorem states that a topological gluing can be realized geometrically by a convex polyhedron precisely if the gluing is topologically a sphere, and at most 360° of material is glued to any one point—that is, every point should have nonnegative curvature.

Based on this tool, Lubiw and O'Rourke use dynamic programming to develop their algorithms. There are $\Omega(n^2)$ subproblems corresponding to gluing subchains of the polygon, assuming that the two ends of the subchain have already been glued together. These subproblems are additionally parameterized by how much angle of material remains at the point to which the two ends of the chain glue in order to maintain positive curvature. It is this parameterization that forces enumeration of all gluings to take exponential time. But for the decision problem of the existence of any gluing, the remaining angle at the ends only needs to be bounded, and only polynomially many subproblems need to be considered, resulting in an $O(n^3)$ algorithm.

A particularly surprising discovery from this work [LO96] is that the well-known "Latin cross" unfolding of the cube can be folded into exactly five convex polyhedra by edge-to-edge gluing: a doubly covered (flat) quadrangle, an (irregular) tetrahedron, a pentahedron, the cube, and an (irregular) octahedron. See Figure 23 for crease patterns and gluing instructions. These foldings are the subject of a video [DDL+99a].



(a) Doubly covered quadrangle (b) Tetrahedron (c) Pentahedron (d) Cube (e) Octahedron

Figure 23: The five edge-to-edge gluings of the Latin cross [LO96].

4.2.2 Non-Edge-to-Edge Gluings

More recently, Demaine, Demaine, Lubiw, and O'Rourke [DDLO00b, DDLO02] have extended this work in various directions, in particular to non-edge-to-edge gluings.

In contrast to edge-to-edge gluings, any convex polygon can be glued into a continuum of distinct convex polyhedra, making it more difficult for an algorithm to enumerate all gluings of a given polygon. Fortunately, there are only finitely many *combinatorially distinct* gluings of any polygon. For convex polygons, there are only polynomially many combinatorially distinct gluings, and they can be enumerated for a given convex polygon in polynomial time. This result generalizes to any polygon in which there is a constant bound on the sharpest angle. For general nonconvex polygons, there can be exponentially many ($2^{\Theta(n)}$) combinatorially distinct gluings, but only that many. Again this corresponds to an algorithm running in $2^{O(n)}$ time. Because of the exponential worst-case lower bound on the number of combinatorially distinct gluings, we are justified both here and in the enumeration algorithm of [LO96] to spend exponential time. It remains open whether there is an output-sensitive algorithm, whose running time is polynomial in the number of resulting gluings, or in the number of gluings desired by the user. For non-edge-to-edge gluings, it even remains open whether there is a polynomial-time algorithm to decide whether a gluing exists.

The algorithms for enumerating all non-edge-to-edge gluings have been implemented independently by Anna Lubiw (July 2000) and by Koichi Hirata [Hir00] (June 2000). These programs have been applied to the example of the Latin cross. There are surprisingly many more, but still finitely many, non-edge-to-edge gluings: a total of 85 distinct gluings (43 modulo symmetry). A manual reconstruction of the polyhedra resulting from these gluings reveals 23 distinct shapes: the cube, seven different tetrahedra, three different pentahedra, four different hexahedra, six different octahedra, and two flat quadrangles [DDLO00a, DO].

Alexander, Dyson, and O'Rourke [ADO02] performed a case study of all the gluings of the square, reconstructing all the incongruent polyhedra that result. This situation is complicated by the existence of entire continua of gluings and polyhedra. Nonetheless, the entire configuration space of the polyhedra can be characterized, as shown in Figure 24. Although in this case it is connected, there are convex polygons of n vertices whose space of all gluings into polyhedra has $\Omega(n^2)$ connected components [DO]. Although it is almost certain that all of these gluings lead to distinct polyhedra, it seems difficult to establish this property without a method for reconstructing the three-dimensional structure, the topic of the next section.

4.2.3 Constructing Polyhedra

Another intriguing open problem in this area [DDLO02] remains relatively unexplored: Aleksandrov's theorem implies that any valid gluing (homeomorphic to a sphere and having nonnegative curvature everywhere) can be folded into a unique convex polyhedron, but how efficiently can this polyhedron be constructed? The key difficulty here is to determine the dihedral angles of the polyhedron, that is, by how much each crease is folded. Finding a (superset of)

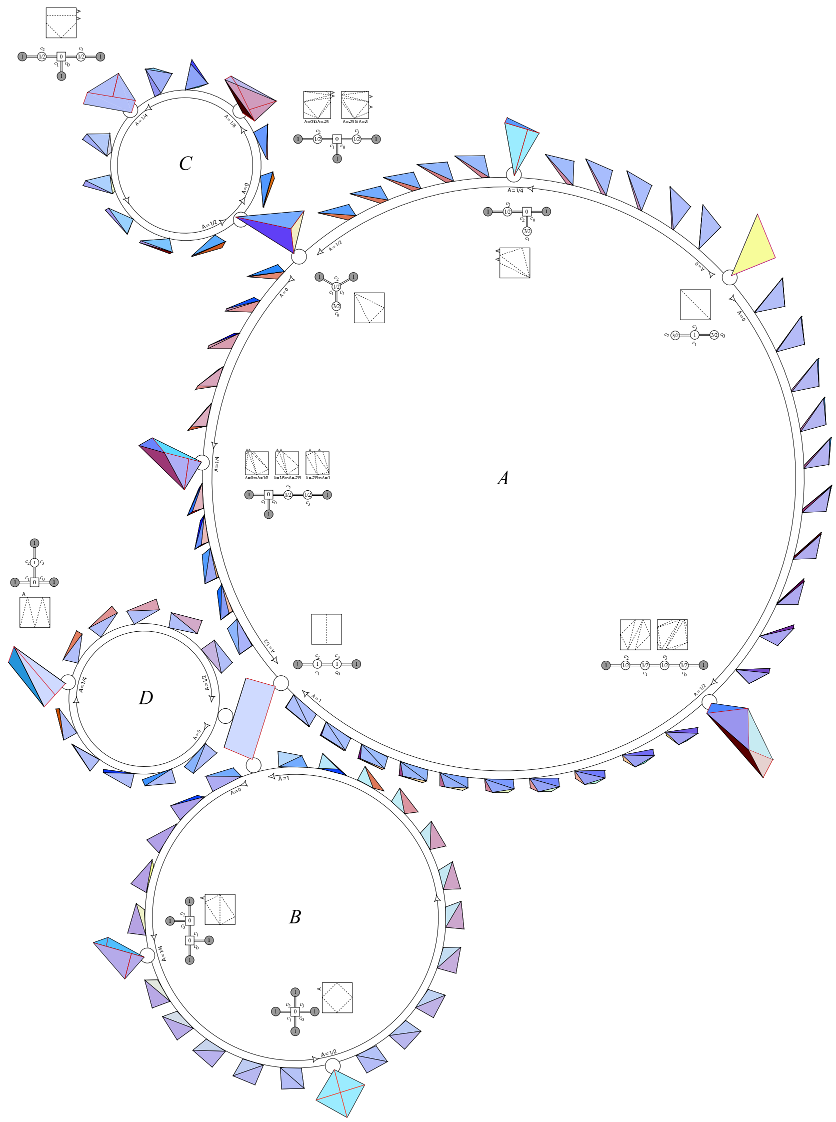


Figure 24: The continua of polyhedra foldable from a square. (Figure 2 of [ADO02].)

the creases is straightforward:[12] every edge of the polyhedron is a shortest path between two positive-curvature vertices, so compute all-pairs shortest paths in the polyhedral metric defined by the gluing [CH96, KO00, Kap99].

[12]Personal communication with Boris Aronov, June 1998. The essence of the argument is also present in Aleksandrov [Ale96].

Sabitov [Sab96] recently presented a finite algorithm for this reconstruction problem, reducing the problem to finding roots of a collection of polynomials of exponentially high degree. The algorithm is based on another his results [Sab98, Sab96] that expresses the volume of a triangulated polyhedron as the root of a polynomial in the edge lengths, independent of how the polyhedron is geometrically embedded in 3-space. (This result was also used to settle the famous Bellows Conjecture [CSW97].) Sabitov's algorithm was recently extended and its bounds improved by Fedorchuk and Pak [FP04] to express the internal vertex-to-vertex diagonal lengths as roots of a polynomial of degree 4^m for a polyhedron of m edges. The polyhedron can easily be reconstructed from these diagonal lengths.

4.3 Folding Nets into Nonconvex Polyhedra

Define a *net* to be a connected edge-to-edge gluing of polygons to form a tree structure, the edges shared by polygons denoting creases. An open problem mentioned in Section 4.2.3 is deciding whether a given net can be folded into a convex polyhedron using only the given creases. More generally, we can ask whether a given net folds into a nonconvex polyhedron. Now Aleksandrov's theorem and Cauchy's rigidity theorem do not apply, so for a given gluing we are no longer easily guaranteed existence or uniqueness.

Given the dihedral angles associated with creases in the net, it is easy to decide foldability in polynomial time [BLS99, Sun99]: we only need to check that edges match up and no two faces cross. Without the dihedral angles, when does a given net fold into any polyhedron? Biedl, Lubiw, and Sun [BLS99, Sun99] proved a closely related problem to be weakly NP-complete: does a given orthogonal net (each face is an orthogonal polygon) fold into an orthogonal polyhedron? The difference with this problem is that it constrains each dihedral angle to be $\pm 90°$. It remained open whether this constraint actually restricted what polyhedra could be folded, even for this particular reduction. More generally, is there a nonorthogonal polyhedron (i.e., one that has at least one dihedral angle not a multiple of $90°$) having orthogonal faces and that is homeomorphic to a sphere? The answer to this question (posed in [BLS99]) turns out to be NO, as proved by Donoso and O'Rourke [DO02]. The answer is YES, however, if the polyhedron is allowed to have genus 6 or larger; on the other hand, the answer remains NO for genus up to 2 [BCD+02]. It remains open whether such nonorthogonal polyhedra with orthogonal faces exist with genus 3, 4, or 5.

4.4 Continuously Folding Polyhedra

The results described so far for polyhedron folding and unfolding are essentially about folded or unfolded states, and not about the continuous process of reaching such states. In the context of paper folding, we saw in Section 3.3.3 that these two notions are largely equivalent. In the context of linkages, we saw that the two notions can differ, particularly in 3D. Relatively little has been studied in the context of polyhedron folding.

One special case that has been explored is orthogonal polyhedra. Specifically, Biedl, Lubiw, and Sun [BLS99, Sun99] have proved that there is an edge-unfolding of an orthogonal polyhedron (which is an orthogonal net) that cannot be folded into the orthogonal polyhedron by a continuous motion that keeps the faces rigid and avoids self-intersection. The basis for their example is the locked polygonal arc in 3D (Figure 10), converted into an orthogonal locked polygonal arc in 3D, and then "thickened" into an orthogonal tube. A single chain of faces in the unfolding is what prevents the continuous foldability.

One would expect, analogous to the results described in Section 3.3.3 [DM01], that collections of polygons hinged together into a tree can be folded into all possible configurations if we allow additional creases during the motion. However, this extension (equivalent to a polygonal piece of paper) remains open. A particularly interesting version of this question, posed in [BLS99], is whether a finite number of additional creases suffice.

An interesting collection of open questions arise when we consider polyhedron foldings with creases only at polyhedron edges. For example, do all convex polyhedra have *continuous* edge-unfoldings? (This question may be easier to answer negatively than the classic edge-unfolding problem.) Figure 25 shows a simple example of such a folding, taken from a longer video [DDL$^+$99a], based on the simple rule of affinely interpolating each dihedral angle from start to finish. Connelly, as reported in [MP03b], asked whether the source unfolding can be *continuously bloomed*, i.e., unfolded so that all dihedral angles increase monotonically. Although an affirmative answer to this question has just been obtained[13], it remains open whether every general unfolding can be executed continuously.

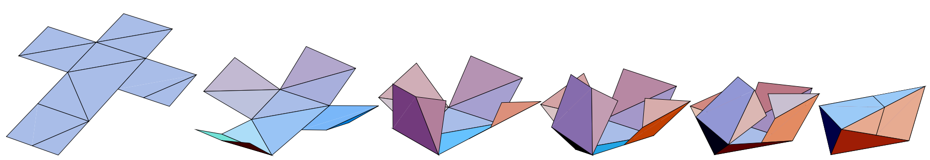


Figure 25: Folding the Latin cross into an octahedron, according to the crease pattern in Figure 23(e), by affinely interpolating all dihedral angles. (Figure 2 of [DDL$^+$99a].)

5 Conclusion and Higher Dimensions

Our goal has been to survey the results in the newly developing area of folding and unfolding, which offers many beautiful mathematical and computational problems. Much progress has been made recently in this area, but many important problems remain open. For example, most aspects of unfolding polyhedra remain unsolved, and we highlight two key problems in this context: can all con-

[13] Personal communication with Stefan Langerman et al., February 2004.

vex polyhedra be edge-unfolded, and can all polyhedra be generally unfolded? Another exciting new direction is the developing connection between linkage folding and protein folding.

Finally, higher dimensions are just beginning to be explored. We mentioned in Section 2.4 that 1D (one-dimensional) linkages in higher dimensions have been explored. But 2D "linkages" in 4D—and higher-dimensional analogs—have received less attention. One model is 2D polygons hinged together at their edges to form a chain. Such a hinged chain has fewer degrees of freedom than a 1D linkage in 3D; for example, a hinged chain can be forced to fold like a planar linkage by extruding the linkage orthogonal to the plane. See Figure 26. Biedl,

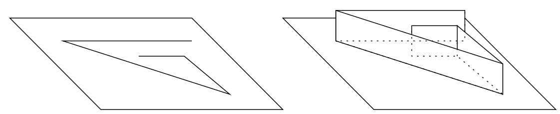


Figure 26: Extruding a linkage into an equivalent collection of polygons (rectangles) hinged together at their edges.

Lubiw, and Sun [BLS99, Sun99] showed that even hinged chains of rectangles do not have connected configuration spaces, by considering an orthogonal version of Figure 10. It would be interesting to explore these chains of rectangles in 4D.

Turning to the origami context, one natural open problem is a generalization of the fold-and-cut problem: given a polyhedral complex drawn on a d-dimensional piece of paper, is it always possible to fold the paper flat (into d-space) while mapping the $(d-1)$-dimensional facets of the complex to a common $(d-1)$-dimensional hyperplane? What if our goal is to map all k-dimensional faces to a common k-dimensional flat, for all $k = 0, 1, \ldots, d$?

Salvador Dali's famous painting ("Christ") of Christ on an unfolded 4D hypercube suggests the possibilities for unfolding higher-dimensional polyhedra. All of the unsolved problems related to unfolding 3D to 2D are equally unsolved in their higher-dimensional analogs. We mentioned in Section 4.1 a rare exception: the vertex-unfolding algorithm generalizes to unfold simplicial manifolds without overlap in arbitrary dimensions. It has recently been established by Miller and Pak [MP03b] that the source unfolding generalizes to higher dimensions to yield nonoverlapping unfoldings, but that the most natural generalization of the star unfolding does not even suffice to unfold, let alone without overlap. Nevertheless, with one general unfolding available, the natural analog of the edge-unfolding question remains: Does every convex d-polytope have a *ridge unfolding*, a cutting of $(d-2)$-dimensional faces that unfolds the polytope into $\mathbb{R}^{d-1}$ without overlap?

Acknowledgements. We appreciate the helpful remarks of Joseph Mitchell, and the suggestions of the referees.

References

[AAOS97] Pankaj K. Agarwal, Boris Aronov, Joseph O'Rourke, and Catherine A. Schevon. Star unfolding of a polytope with applications. *SIAM Journal on Computing*, 26(6):1689–1713, December 1997.

[ABC+00] Hee-Kap Ahn, Prosenjit Bose, Jurek Czyzowicz, Nicolas Hanusse, Evangelos Kranakis, and Pat Morin. Flipping your lid. *Geombinatorics*, 10(2):57–63, 2000.

[ABD+03] Oswin Aichholzer, David Bremner, Erik D. Demaine, Henk Meijer, Vera Sacristán, and Michael Soss. Long proteins with unique optimal foldings in the H-P model. *Computational Geometry: Theory and Applications*, 25(1–2):139–159, May 2003.

[ABD+04] Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Martin L. Demaine, Joseph S. B. Mitchell, Saurabh Sethia, and Steven S. Skiena. When can you fold a map? *Computational Geometry: Theory and Applications*, 29(1):23–46, September 2004.

[ACD+02] Oswin Aichholzer, Carmen Cortés, Erik D. Demaine, Vida Dujmović, Jeff Erickson, Henk Meijer, Mark Overmars, Belén Palop, Suneeta Ramaswami, and Godfried T. Toussaint. Flipturning polygons. *Discrete & Computational Geometry*, 28(2):231–253, August 2002.

[ADD+02] Greg Aloupis, Erik D. Demaine, Vida Dujmović, Jeff Erickson, Stefan Langerman, Henk Meijer, Ileana Streinu, Joseph O'Rourke, Mark Overmars, Michael Soss, and Godfried T. Toussaint. Flat-state connectivity of linkages under dihedral motions. In *Proceedings of the 13th Annual International Symposium on Algorithms and Computation*, volume 2518 of *Lecture Notes in Computer Science*, pages 369–380, Vancouver, Canada, November 2002.

[ADE+01] Oswin Aichholzer, Erik D. Demaine, Jeff Erickson, Ferran Hurtado, Mark Overmars, Michael A. Soss, and Godfried T. Toussaint. Reconfiguring convex polygons. *Computational Geometry: Theory and Applications*, 20(1–2):85–95, October 2001.

[ADM+02] Greg Aloupis, Erik D. Demaine, Henk Meijer, Joseph O'Rourke, Ileana Streinu, and Godfried Toussaint. Flat-state connectedness of fixed-angle chains: Special acute chains. In *Proceedings of the 14th Canadian Conference on Computational Geometry*, pages 27–30, Lethbridge, Canada, August 2002.

[ADO02] Rebecca Alexander, Heather Dyson, and Joseph O'Rourke. The convex polyhedra foldable from a square. In *Proceedings of the Japan Conference on Discrete and Computational Geometry*, pages 31–32, Tokyo, 2002.

[AFM03] Esther M. Arkin, Sándor P. Fekete, and Joseph S. B. Mitchell. An algorithmic study of manufacturing paperclips and other folded structures. *Computational Geometry: Theory and Applications*, 25(1–2):117–138, 2003.

[AGP02] Boris Aronov, Jacob E. Goodman, and Richard Pollack. Convexification of planar polygons in $\mathbb{R}^3$. Manuscript, June 2002. Revision of October 1999 draft.

[AKRW04] Helmut Alt, Christian Knauer, Günter Rote, and Sue Whitesides. On the complexity of the linkage reconfiguration problem. In J. Pach, editor, *Towards a Theory of Geometric Graphs*, volume 342 of *Contemporary Mathematics*, pages 1–14. American Mathematical Society, 2004.

[Ale50] A. D. Aleksandrov. *Convex Polyhedra* (in Russian). State Press of Technical and Theoretical Literature, Moscow, 1950. See also the German translation, *Konvexe Polyeder*, Akademie Verlag, 1958.

[Ale96] A. D. Aleksandrov. Existence of a convex polyhedron and a convex surface with a given metric. In Yu. G. Reshetnyak and S. S. Kutateladze, editors, *A. D. Aleksandrov: Selected Works: Part I*, pages 169–173. Gordon and Breach, Australia, 1996. Translation of *Doklady Akad. Nauk SSSR, Matematika*, Vol. 30, No. 2, 103–106 (1941).

[AO92] Boris Aronov and Joseph O'Rourke. Nonoverlap of the star unfolding. *Discrete & Computational Geometry*, 8(3):219–250, 1992.

[BCD+02] Therese Biedl, Timothy M. Chan, Erik D. Demaine, Martin L. Demaine, Paul Nijjar, Ryuhei Uehara, and Ming wei Wang. Tighter bounds on the genus of nonorthogonal polyhedra built from rectangles. In *Proceedings of the 14th Canadian Conference on Computational Geometry*, pages 105–108, Lethbridge, Canada, August 2002.

[BDD+98] Therese Biedl, Erik Demaine, Martin Demaine, Anna Lubiw, Mark Overmars, Joseph O'Rourke, Steve Robbins, and Sue Whitesides. Unfolding some classes of orthogonal polyhedra. In *Proceedings of the 10th Canadian Conference on Computational Geometry*, Montréal, Canada, August 1998. http://cgm.cs.mcgill.ca/cccg98/proceedings/cccg98-biedl-unfolding.ps.gz%.

[BDD+01] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides. Locked and unlocked polygonal chains in three dimensions. *Discrete & Computational Geometry*, 26(3):269–281, October 2001. The full version is Technical Report 060, Smith College, 1999, and arXiv:cs.CG/9910009, http://www.arXiv.org/abs/cs.CG/9910009.

[BDD+02] Therese Biedl, Erik Demaine, Martin Demaine, Sylvain Lazard, Anna Lubiw, Joseph O'Rourke, Steve Robbins, Ileana Streinu, Godfried Toussaint, and Sue Whitesides. A note on reconfiguring tree linkages: Trees can lock. *Discrete Applied Mathematics*, 117(1–3):293–297, 2002. The full version is Technical Report SOCS-00.7, School of Computer Science, McGill University, September 2000, and Computing Research Repository paper cs.CG/9910024. http://www.arXiv.org/abs/cs.CG/9910024.

[BDE+03] Marshall Bern, Erik D. Demaine, David Eppstein, Eric Kuo, Andrea Mantler, and Jack Snoeyink. Ununfoldable polyhedra with convex faces. *Computational Geometry: Theory and Applications*, 24(2):51–62, February 2003.

[BDEH01] Marshall Bern, Erik Demaine, David Eppstein, and Barry Hayes. A disk-packing algorithm for an origami magic trick. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, pages 17–28, Monterey, California, March 2001. Improvement of version appearing in *Proceedings of the International Conference on Fun with Algorithms*, Isola d'Elba, Italy, June 1998, pages 32–42.

[BGRT99] Prosenjit Bose, Francisco Gomez, Pedro Ramos, and Godfried Toussaint. Drawing nice projections of objects in space. *Journal of Visual Communication and Image Representation*, 10(2):155–172, 1999.

[BH96] Marshall Bern and Barry Hayes. The complexity of flat origami. In *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 175–183, Atlanta, January 1996.

[Bie05] Therese Biedl. Polygons needing many flipturns. *Discrete & Computational Geometry*, 2005. To appear.

[BL98] Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic (*HP*) model is NP-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[BLS99] Therese Biedl, Anna Lubiw, and Julie Sun. When can a net fold to a polyhedron? In *Proceedings of the 11th Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999. http://www.cs.ubc.ca/conferences/CCCG/elec_proc/fp16.ps.gz.

[Cau13] A. L. Cauchy. Deuxième mémoire sur les polygones et polyèdres (Second memoire on polygons and polyhedra). *Journal de l'École Polytechnique*, 9:87–98, May 1813.

[CD90] Hue Sun Chan and Ken A. Dill. Origins of structure in globular proteins. *Proceedings of the National Academy of Sciences USA*, 87:6388–6392, August 1990.

[CD91] Hue Sun Chan and Ken A. Dill. Polymer principles in protein structure and stability. *Annual Reviews of Biophysics and Biophysical Chemistry*, 20:447–490, 1991.

[CD93] Hue Sun Chan and Ken A. Dill. The protein folding problem. *Physics Today*, pages 24–32, February 1993.

[CD04] Robert Connelly and Erik D. Demaine. Geometry and topology of polygonal linkages. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 9, pages 197–218. CRC Press, 2nd edition, 2004.

[CDIO04] Jason H. Cantarella, Erik D. Demaine, Hayley N. Iben, and James F. O'Brien. An energy-driven approach to linkage unfolding. In *Proceedings of the 20th Annual ACM Symposium on Computational Geometry*, pages 134–143, Brooklyn, New York, June 2004.

[CDR02] Robert Connelly, Erik D. Demaine, and Günter 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 3-space*, pages 287–311. American Mathematical Society, 2002.

[CDR03] Robert Connelly, Erik D. Demaine, and Günter Rote. Straightening polygonal arcs and convexifying polygonal cycles. *Discrete & Computational Geometry*, 30(2):205–239, September 2003.

[CGP+98] Pierluigi Crescenzi, Deborah Goldman, Christos Papadimitriou, Antonio Piccolboni, and Mihalis Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 5(3), 1998.

[CH96] Jindong Chen and Yijie Han. Shortest paths on a polyhedron. *International Journal of Computational Geometry and Applications*, 6(2):127–144, 1996.

[CJ98] Jason Cantarella and Heather Johnston. Nontrivial embeddings of polygonal intervals and unknots in 3-space. *Journal of Knot Theory and Its Ramifications*, 7(8):1027–1039, 1998.

[CKM+01] Jorge Alberto Calvo, Danny Krizanc, Pat Morin, Michael Soss, and Godfried Toussaint. Convexifying polygons with simple projections. *Information Processing Letters*, 80(2):81–86, 2001.

[CO01] Roxana Cocan and Joseph O'Rourke. Polygonal chains cannot lock in 4D. *Discrete & Computational Geometry*, 20(3):105–129, November 2001.

[Con80] Robert Connelly. The rigidity of certain cabled frameworks and the second-order rigidity of arbitrarily triangulated convex surfaces. *Advances in Mathematics*, 37(3):272–299, 1980.

[Con93] Robert Connelly. Rigidity. In *Handbook of Convex Geometry*, volume A, pages 223–271. North-Holland, Amsterdam, 1993.

[Cro97] Peter R. Cromwell. Equality, rigidity, and flexibility. In *Polyhedra*, chapter 6, pages 219–247. Cambridge University Press, 1997.

[CSW97] R. Connelly, I. Sabitov, and A. Walz. The bellows conjecture. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 38(1):1–10, 1997.

[DD97] Erik D. Demaine and Martin L. Demaine. Computing extreme origami bases. Technical Report CS-97-22, Department of Computer Science, University of Waterloo, Waterloo, Canada, May 1997.

[DD01] Erik D. Demaine and Martin L. Demaine. Recent results in computational origami. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, pages 3–16, Monterey, California, March 2001.

[DDL98] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Folding and cutting paper. In J. Akiyama, M. Kano, and M. Urabe, editors, *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Computer Science*, pages 104–117, Tokyo, Japan, December 1998.

[DDL+99a] Erik Demaine, Martin Demaine, Anna Lubiw, Joseph O'Rourke, and Irena Pashchenko. Metamorphosis of the cube. In *8th Annual Video Review of Computational Geometry, Proceedings of the 15th Annual ACM Symposium on Computational Geometry*, pages 409–410, Miami Beach, Florida, June 1999. http://theory.csail.mit.edu/~edemaine/metamorphosis/.

[DDL99b] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Folding and one straight cut suffice. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 891–892, Baltimore, MD, January 1999.

[DDLO00a] Erik Demaine, Martin Demaine, Anna Lubiw, and Joseph O'Rourke. The 85 foldings of the Latin cross, 2000. http://theory.csail.mit.edu/~edemaine/aleksandrov/cross/.

[DDLO00b] Erik Demaine, Martin Demaine, Anna Lubiw, and Joseph O'Rourke. Examples, counterexamples, and enumeration results for foldings and unfoldings between polygons and polytopes. Technical Report 069, Smith College, Northampton, MA, July 2000. http://www.arXiv.org/abs/cs.CG/0007019.

[DDLO02] Erik D. Demaine, Martin L. Demaine, Anna Lubiw, and Joseph O'Rourke. Enumerating foldings and unfoldings between polygons and polytopes. *Graphs and Combinatorics*, 18(1):93–104, 2002. http://www.arXiv.org/abs/cs.CG/0107024.

[DDM99] Erik D. Demaine, Martin L. Demaine, and Joseph S. B. Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. In *Proceedings of the 15th Annual ACM Symposium on Computational Geometry*, Miami Beach, Florida, June 1999.

[DDM00] Erik D. Demaine, Martin L. Demaine, and Joseph S. B. Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. *Computational Geometry: Theory and Applications*, 16(1):3–21, 2000.

[DDMO04] Erik D. Demaine, Satyan L. Devadoss, Joseph S. B. Mitchell, and Joseph O'Rourke. Continuous foldability of polygonal paper. In *Proceedings of the 16th Canadian Conference on Computational Geometry*, pages 64–67, Montréal, Canada, August 2004.

[DEE+01] Erik D. Demaine, David Eppstein, Jeff Erickson, George W. Hart, and Joseph O'Rourke. Vertex-unfoldings of simplicial manifolds. arXiv:cs.CG/0110054, October 2001. http://www.arXiv.org/abs/cs.CG/0110054.

[Dem00] Erik D. Demaine. Folding and unfolding linkages, paper, and polyhedra. In *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 2098 of *Lecture Notes in Computer Science*, pages 113–124, Tokyo, Japan, November 2000.

[Dil90] Ken A. Dill. Dominant forces in protein folding. *Biochemistry*, 29(31):7133–7155, August 1990.

[DLO03] Erik D. Demaine, Stefan Langerman, and Joseph O'Rourke. Geometric restrictions on producible polygonal protein chains. In *Proceedings of the 14th Annual International Symposium on Algorithms and Computation*, volume 2906 of *Lecture Notes in Computer Science*, pages 395–404, Kyoto, Japan, December 2003. To appear in *Algorithmica*.

[DLOS02] Erik D. Demaine, Stefan Langerman, Joseph O'Rourke, and Jack Snoeyink. Interlocked open linkages with few joints. In *Proceedings of the 18th Annual ACM Symposium on Computational Geometry*, pages 189–198, Barcelona, Spain, June 2002.

[DLOS03] Erik D. Demaine, Stefan Langerman, Joseph O'Rourke, and Jack Snoeyink. Interlocked open and closed linkages with few joints. *Computational Geometry: Theory and Applications*, 26(1):37–45, August 2003.

[DM01] Erik D. Demaine and Joseph S. B. Mitchell. Reaching folded states of a rectangular piece of paper. In *Proceedings of the 13th Canadian Conference on Computational Geometry*, pages 73–75, Waterloo, Canada, August 2001. http://compgeo.math.uwaterloo.ca/~cccg01/proceedings/short/eddemaine-33029.ps.

[DO] Erik D. Demaine and Joseph O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, and Polyhedra*. Cambridge University Press. In preparation. http://www.fucg.org.

[DO01] Erik D. Demaine and Joseph O'Rourke. Open problems from CCCG 2000. In *Proceedings of the 13th Canadian Conference on Computational Geometry*, Waterloo, Canada, August 2001. http://compgeo.math.uwaterloo.ca/~cccg01/proceedings/short/eddemaine-18187.ps.gz.

[DO02] Melody Donoso and Joseph O'Rourke. Nonorthogonal polyhedra built from rectangles, August 2002. http://www.arXiv.org/abs/cs.CG/0110059.

[Dür77] Albrecht Dürer. *The Painter's Manual: A Manual of Measurement of Lines, Areas, and Solids by Means of Compass and Ruler Assembled by Albrecht Dürer for the Use of All Lovers of Art with Appropriate Illustrations Arranged to be Printed in the Year MDXXV*. Abaris Books, Inc., New York, 1977. English translation of *Unterweysung der Messung mit dem Zirkel un Richtscheyt in Linien Ebnen und Gantzen Corporen*, 1525.

[Erd35] Paul Erdős. Problem 3763. *American Mathematical Monthly*, 42:627, 1935.

[FHM+01] Thomas Fevens, Antonio Hernandez, Antonio Mesa, Patrick Morin, Michael Soss, and Godfried Toussaint. Simple polygons with an infinite sequence of deflations. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 42(2):307–311, 2001.

[FP04] Maksym Fedorchuk and Igor Pak. Rigidity and polynomial invariants of convex polytopes. Manuscript, July 2004. http://www-math.mit.edu/~pak/pp16.pdf.

[Gar95] Martin Gardner. Paper cutting. In *New Mathematical Diversions (Revised Edition)*, chapter 5, pages 58–69. The Mathematical Association of America, Washington, D.C., 1995. Appeared in *Scientific American*, June 1960.

[GLO+04] Julie Glass, Stefan Langerman, Joseph O'Rourke, Jack Snoeyink, and Jianyuan K. Zhong. A 2-chain can interlock with a k-chain. arxiv:cs.CG/0410052, October 2004. http://arxiv.org/abs/cs.CG/0410052.

[Grü95] Branko Grünbaum. How to convexify a polygon. *Geombinatorics*, 5:24–30, July 1995.

[GZ98] Branko Grünbaum and Joseph Zaks. Convexification of polygons by flips and by flipturns. Technical Report 6/4/98, Department of Mathematics, University of Washington, Seattle, 1998.

[Har73] National standards and emblems. *Harper's New Monthly Magazine*, 47(278):171–181, July 1873.

[Hay98] Brian Hayes. Prototeins. *American Scientist*, 86:216–221, 1998.

[HI96] William E. Hart and Sorin Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal. *Journal of Computational Biology*, 3(1):53–96, Spring 1996.

[Hir00] Koichi Hirata. Find all polytope, 2000. http://weyl.ed.ehime-u.ac.jp/cgi-bin/WebObjects/Polytope2.

[HJW84] John Hopcroft, Deborah Joseph, and Sue Whitesides. Movement problems for 2-dimensional linkages. *SIAM Journal on Computing*, 13(3):610–629, August 1984.

[Hou22] Harry Houdini. *Paper Magic*, pages 176–177. E. P. Dutton & Company, 1922. Reprinted by Magico Magazine.

[Hul94] Thomas Hull. On the mathematics of flat origamis. *Congressus Numerantium*, 100:215–224, 1994.

[Hul03] Thomas Hull. Counting mountain-valley assignments for flat folds. *Ars Combinatoria*, 67:175–187, April 2003.

[JS99] D. Jordan and M. Steiner. Configuration spaces of mechanical linkages. *Discrete & Computational Geometry*, 22:297–315, 1999.

[Jus94] Jacques Justin. Towards a mathematical theory of origami. In Koryo Miura, editor, *Proceedings of the 2nd International Meeting of Origami Science and Scientific Origami*, pages 15–29, Otsu, Japan, November–December 1994.

[Kap99] Sanjiv Kapoor. Efficient computation of geodesic shortest paths. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 770–779, Atlanta, Georgia, May 1999.

[Kaw89] Toshikazu 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.

[Kem76] A. B. Kempe. On a general method of describing plane curves of the n^{th} degree by linkwork. *Proceedings of the London Mathematical Society*, 7:213–216, 1876.

[KM95] Michael Kapovich and John Millson. On the moduli space of polygons in the Euclidean plane. *Journal of Differential Geometry*, 42(1):133–164, 1995.

[KM02] Michael Kapovich and John J. Millson. Universality theorems for configuration spaces of planar linkages. *Topology*, 41(6):1051–1107, 2002.

[KO00] Biliana Kaneva and Joseph O'Rourke. An implementation of Chen & Han's shortest paths algorithm. In *Proceedings of the 12th Canadian Conference on Computational Geometry*, pages 139–146, Fredericton, Canada, August 2000. http://www.cs.unb.ca/conf/cccg/eProceedings/8.ps.gz.

[Lan94a] Robert J. Lang. Mathematical algorithms for origami design. *Symmetry: Culture and Science*, 5(2):115–152, 1994.

[Lan94b] Robert J. Lang. The tree method of origami design. In Koryo Miura, editor, *Proceedings of the 2nd International Meeting of Origami Science and Scientific Origami*, pages 73–82, Otsu, Japan, November–December 1994.

[Lan96] Robert J. Lang. A computational algorithm for origami design. In *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, pages 98–105, Philadelphia, PA, May 1996.

[Lan98] Robert J. Lang. *TreeMaker 4.0: A Program for Origami Design*, 1998. http://origami.kvi.nl/programs/treemaker/trmkr40.pdf.

[Lan03] Robert J. Lang. *Origami Design Secrets: Mathematical Methods for an Ancient Art*. A K Peters, 2003.

[LD89] K. F. Lau and K. A. Dill. A lattice statistical mechanics model of the conformation and sequence spaces of proteins. *Macromolecules*, 22:3986–3997, 1989.

[LD90] K. F. Lau and K. A. Dill. Theory for protein mutability and biogenesis. *Proceedings of the National Academy of Sciences USA*, 87:638–642, January 1990.

[LO96] Anna Lubiw and Joseph O'Rourke. When can a polygon fold to a polytope? Technical Report 048, Smith College, June 1996.

[Lun68] W. F. Lunnon. A map-folding problem. *Mathematics of Computation*, 22(101):193–199, 1968.

[Lun71] W. F. Lunnon. Multi-dimensional map-folding. *The Computer Journal*, 14(1):75–80, February 1971.

[LW91] David J. Lipman and W. John Wilber. Modelling neutral and selective evolution of protein folding. *Proceedings of The Royal Society of London, Series B*, 245(1312):7–11, 1991.

[LW95] W. J. Lenhart and S. H. Whitesides. Reconfiguring closed polygonal chains in Euclidean d-space. *Discrete & Computational Geometry*, 13:123–140, 1995.

[ML94] Kenneth M. Merz and Scott M. Le Grand, editors. *The Protein Folding Problem and Tertiary Structure Prediction*. Birkhäuser, Boston, 1994.

[MMP87] Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, August 1987.

[Mon91] John Montroll. *African Animals in Origami*. Dover Publications, 1991.

[Mon93] John Montroll. *Origami Inside-Out*. Dover Publications, 1993.

[MP03a] Ezra Miller and Igor Pak. Metric combinatorics of convex polyhedra: cut loci and nonoverlapping unfoldings. Manuscript, December 2003. http://www.math.umn.edu/~ezra/Foldout/fold.pdf.

[MP03b] Ezra Miller and Igor Pak. Metric combinatorics of convex polyhedra: Cut loci and nonoverlapping unfoldings. http://www-math.mit.edu/~pak/fold84.pdf, 2003.

[Nag39] Béla de Sz. Nagy. Solution to problem 3763. *American Mathematical Monthly*, 46:176–177, March 1939.

[O'R98] Joseph O'Rourke. Folding and unfolding in computational geometry. In *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Computer Science*, pages 258–266, Tokyo, Japan, December 1998.

[O'R02] Joseph O'Rourke. Computational geometry column 43. *International Journal of Computational Geometry and Applications*, 12(3):263–265, 2002. Also in SIGACT News, 33(1):58-60 (2002), Issue 122.

[Rot03] Günter Rote. Pseudotriangulations—A survey and recent results, October 2003. Course notes, Journées de Géométrie Algorithmique 2003. http://page.mi.fu-berlin.de/~rote/Papers/slides/Pseudotriangulations-JGA2003/Pseudotriangulations.pdf.

[Sab96] I. Kh. Sabitov. The volume of a polyhedron as a function of its metric and algorithmical solution of the main problems in the metric theory of polyhedra. In *Internat. School–Seminar Devoted to the N. V. Efimov's Memory*, pages 64–65. Rostov University, 1996.

[Sab98] I. Kh. Sabitov. The volume as a metric invariant of polyhedra. *Discrete & Computational Geometry*, 20(4):405–425, 1998.

[Sal73] G. T. Sallee. Stretching chords of space curves. *Geometriae Dedicata*, 2:311–315, 1973.

[Sch87] Catherine Schevon. Unfolding polyhedra. sci.math Usenet article, February 1987. See http://www.ics.uci.edu/~eppstein/gina/unfold.html.

[Sch89] Catherine Schevon. *Algorithms for Geodesics on Polytopes*. PhD thesis, Johns Hopkins University, 1989.

[Sen21] Kan Chu Sen. *Wakoku Chiyekurabe (Mathematical Contests)*. 1721. Excerpts available from http://theory.csail.mit.edu/~edemaine/foldcut/sen_book.html.

[SEO03] Michael Soss, Jeff Erickson, and Mark Overmars. Preprocessing chains for fast dihedral rotations is hard or even impossible. *Computational Geometry: Theory and Applications*, 26(3):235–246, 2003.

[She75] G. C. Shephard. Convex polytopes with convex nets. *Mathematical Proceedings of the Cambridge Philosophical Society*, 78:389–403, 1975.

[Smi76] John S. Smith. Origami profiles. *British Origami*, 58, June 1976.

[Smi93] John S. Smith. *Pureland Origami 1, 2, and 3*. British Origami Society. Booklets 14, 29, and 43, 1980, 1988, and 1993.

[Sos01] Michael Soss. *Geometric and Computational Aspects of Molecular Reconfiguration*. PhD thesis, School of Computer Science, McGill University, 2001.

[ST00] Michael Soss and Godfried T. Toussaint. Geometric and computational aspects of polymer reconfiguration. *Journal of Mathematical Chemistry*, 27(4):303–318, 2000.

[Str00] Ileana Streinu. A combinatorial approach to planar non-colliding robot arm motion planning. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 443–453, Redondo Beach, California, November 2000.

[Sun99] Julie Sun. Folding orthogonal polyhedra. Master's thesis, Department of Computer Science, University of Waterloo, 1999.

[Tou99a] Godfried Toussaint. Computational polygonal entanglement theory. In *Proceedings of the VIII Encuentros de Geometria Computacional*, pages 269–278, Castellon, Spain, July 1999.

[Tou99b] Godfried Toussaint. The Erdős-Nagy theorem and its ramifications. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999. http://www.cs.ubc.ca/conferences/CCCG/elec_proc/fp19.ps.gz.

[Tou01] Godfried Toussaint. A new class of stuck unknots in Pol_6. *Beiträge zur Algebra und Geometrie (Contributions to Algebra and Geometry)*, 42(2):1027–1039, 2001.

[UM93a] Ron Unger and John Moult. Finding the lowest free energy conformation of a protein is a NP-hard problem: Proof and implications. *Bulletin of Mathematical Biology*, 55(6):1183–1198, 1993.

[UM93b] Ron Unger and John Moult. A genetic algorithm for 3D protein folding simulations. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 581–588, San Mateo, CA, 1993.

[UM93c] Ron Unger and John Moult. Genetic algorithms for protein folding simulations. *Journal of Molecular Biology*, 231(1):75–81, 1993.

[Wan97] Cheng-Hua Wang. *Manufacturability-driven decomposition of sheet metal*. PhD thesis, Carnegie Mellon University, Robotics Institute, September 1997. Technical report CMU-RI-TR-97-35.

[Weg93] Bernd Wegner. Partial inflation of closed polygons in the plane. *Beiträge zur Algebra und Geometrie*, 34(1):77–85, 1993.

[Whi92] Sue Whitesides. Algorithmic issues in the geometry of planar linkage movement. *Australian Computer Journal*, 24(2):42–50, May 1992.