Recent Results in Computational Origami

Erik D. Demaine* Martin L. Demaine*

Abstract

Computational origami is a recent branch of computer science studying efficient algorithms for solving paper-folding problems. This field essentially began with Robert Lang's work on algorithmic origami design [25], starting around 1993. Since then, the field of computational origami has grown significantly. The purpose of this paper is to survey the work in the field, with a focus on recent results, and to present several open problems that remain. The survey cannot hope to be complete, but we attempt to cover most areas of interest.

1 Overview

Most results in computational origami fit into at least one of three categories: universality results, efficient decision algorithms, and computational intractability results.

A *universality result* shows that, subject to a certain model of folding, everything is possible. For example, any tree-shaped origami base (Section 2.1), any polygonal silhouette (Section 2.3), and any polyhedral surface (Section 2.3) can be folded out of a large-enough 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 3.4).

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 any flat origami (Section 3.2), even when folds are restricted to simple folds (Section 3.4). These results mean that there are no polynomial-time algorithms for these problems, unless some of the hardest computational problems 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.* Basically, in origami design, some aspects of the target configuration are specified, and the goal is to design a suitable target that can be folded out of paper. In

*Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, {`eddemaine`, `mldemaine`}`@uwaterloo.ca`

origami foldability, the target configuration is unspecified and arbitrary, and rather the initial configuration is specified, specifically 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 an understanding origami foldability is a necessary component to origami design, the results indicate that in fact origami design is much easier to solve than origami foldability which is usually intractable.

Our survey of computational origami is partitioned accordingly into Section 2 (origami design) and Section 3 (origami foldability). In addition, Section 4 briefly surveys some of the other problems in a more general field of geometry called *folding and unfolding*. Specifically, we describe a one-dimensional linkage-folding problem (Section 4.1) and unfolding polyhedra into nonoverlapping nets (Section 4.2).

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. Most closely related to "traditional" origami design is Lang's TreeMaker work (Section 2.1), 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 2.2), which can be used for a magic trick involving folding and one complete straight cut. Another approach is to design an origami with a specific silhouette or three-dimensional shape (Section 2.3), although the algorithms developed so far do not lead to practical foldings. A recent specific type of origami is an origami tessellation (Section 2.4), which "folds" a tiling of the plane. A nonstandard form of origami is to start with a piece of paper that is not flat, but rather the surface of a polyhedron, and the goal is to flatten that surface (Section 2.5).

2.1 TreeMaker

TreeMaker is a computer program by Robert Lang that implements the *tree method* 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 [24]. Here we concentrate on Lang's work [23, 24, 25]; over the past several years, he has developed the tree method to the point where an algorithm and computer program have been explicitly defined and implemented.

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. More precisely, the tree method designs a *uniaxial base* [23], which must have the following properties: the base lies above the xy plane, all faces of paper 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 tree. The theorem is that every geometric tree (unrooted tree with cyclicly ordered children and prescribed edge lengths) is the projection of a uniaxial base, e.g., folded from a square.

The crease pattern for such a base can be found efficiently, certainly in $O(N \operatorname{polylog} N)$ time where N is the number of creases (the output size). It is unclear whether the number of creases required is bounded in terms of the combinatorial complexity of the input, i.e., the number of vertices in the input tree. Optimizing the base to make maximal use of the paper is a difficult nonlinear constrained optimization problem, but the TreeMaker software has shown the viability of existing methods for finding good local optima. Indeed, additional practical constraints can be imposed, such as symmetry in the crease pattern, or that angles of creases are integer multiples of some value (e.g., 22.5°) subject to some flexibility in the edge lengths.

TreeMaker finds a crease pattern that results in the desired uniaxial base. Work on a related problem (see the next section) by Bern, Demaine, Eppstein, and Hayes [4] suggests a method for finding an appropriate mountain-valley assignment for the crease pattern, and possibly also the resulting folded state.

The reader is referred to [25] in this proceedings for more details about the tree method and TreeMaker.

2.2 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 [16], but has a much longer history, going as far back as 1721; see [9].

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 solution [9, 10] is based on a structure called the straight skeleton, which captures the symmetries of the graph, thereby exploiting a more global structure to the problem. The second solution [4] is based on disk packing to make the problem more local, and achieves efficient bounds on the number of creases. See Figure 1 for examples of the first method, and [4] in this proceedings for an example and more information on the second method.

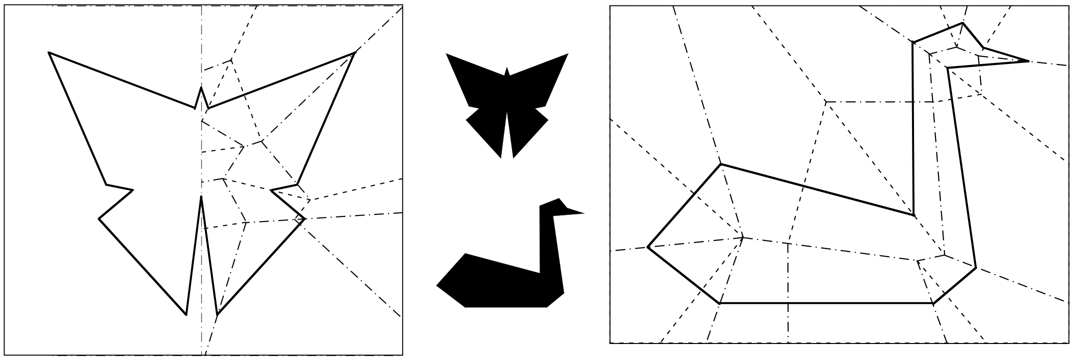

Figure 1: Crease patterns for folding a rectangle of paper flat so that one complete straight cut makes a butterfly (left) or a swan (right), based on [9, 10].

While this problem may not seem directly connected to pure folding because of the one cut, the equivalent problem of folding to line up a collection of edges is in fact closely connected to origami design. Specifically, one subproblem that arises in TreeMaker 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*.

2.3 Silhouettes and Polyhedra

A more direct approach to origami design is to impose 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 (shapes made up of flat sides) is this possible? This problem is implicit throughout origami design, and was first formally posed by Bern and Hayes in 1996 [5]. The surprising answer is "always," as established by Demaine, Demaine, and Mitchell in 1999 [13].

The basic idea of the approach is to fold the piece of paper into a thin strip of paper, 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 [13] for this process. One algorithm maximizes paper usage; the amount of paper required can be made arbitrarily close to the surface area of the shape, which is optimal. 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 [13].

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* [29]. An example is shown in Figure 2.

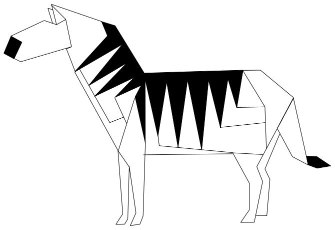

Figure 2: A flat folding of a square of paper, black on one side and white on the other side, designed by John Montroll [28, pp. 94–103].

Of course, because of the reliance on 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 [13] 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 foldings of silhouettes, two-color patterns, and polyhedra.

2.4 Origami Tessellations

Roughly, an origami tessellation is a flat folding of a piece of paper based on a *tessellation* or *tiling* of the plane [17]. One way to make this notion more precise [22, 36] is to consider the whole plane as the piece of paper and define a *symmetric origami tessellation* to be a flat folding of the plane whose symmetry group is one of the 17 crystallographic groups. More generally, an origami tessellation might be defined to be any flat folding of the infinite plane in which no bounded region of the plane contains all of the vertices of the crease pattern. In such a folding, the crease pattern (at least) will necessarily form some kind of tiling.

An origami tessellation can be related to a tiling in various ways. Typically, there are large faces in the flat folded state that remain uncreased, and these faces are precisely the tiles of a tiling. One method for specifying creases around these faces is based on shrinking the tiles and introducing the dual tiling in the resulting gaps, using the notion of a hinged primal-dual tiling (see, e.g., [37]). Bateman [3] has formalized this method to the point of a computer implementation, called Tess. See [3] in this proceedings for more details.

Some of the key people working on origami tessellations include Alex Bateman [3], Paulo Barreto [2, 31], S. Fujimoto [15], Thomas Hull (unpublished), Toshikazu Kawasaki [22], Robert Lang (unpublished), Chris Palmer [31, 32], and Helena Verrill [36]. Unfortunately, much of the work on origami tessellations has not been written formally, so the exact computational results are unclear. Certainly a wide range of origami tessellations have been designed, many using precise algorithms. One of the simplest class of examples are the 11 Archimedian tilings. A characterization of which tilings lead to origami tessellations remains an intriguing open problem.

2.5 Flattening Polyhedra

In most forms of computational origami, the piece of paper is flat: a polygon in the plane, possibly with holes. Suppose instead we start with a piece of paper that is a polyhedral surface in space. How can polyhedra be folded? Specifically, a natural question [11] is whether every polyhedron can be *flattened*: folded into a flat origami. Demaine, Demaine, and Lubiw [11] have shown that there are flattened states of several classes of polyhedra, including convex polyhedra and orthogonal polyhedra. See Figure 3 for an example. Recently (March 2001), together with Barry Hayes, it has been shown that all polyhedra have flattened states.

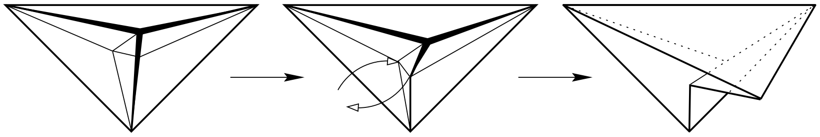

Figure 3: Flattening a tetrahedron, from left to right. Note that the faces are not flat in the middle picture.

3 Origami Foldability

We distinguish *origami foldability* from origami design as starting from 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 that the resulting origami should be flat, i.e., lie in a plane.

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°. This is known as Kawasaki's theorem [5, 18, 20, 21]. When the angle condition is satisfied, a characterization of valid mountain-valley assignments and flat foldings can be found in linear time [5, 20], using Maekawa's theorem [5, 18, 20] and another theorem of Kawasaki [5, 18, 21] 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 [19].

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 [5] proved that consistency can be resolved efficiently when it is possible: local foldability can be tested in linear time.

3.2 Existence of Folded States

Given a crease pattern, does it have a flat folded state? Bern and Hayes [5] 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 [5] 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 Equivalence to Continuous Folding Process

In the previous section and Section 2.5 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 formally, by Demaine and Mitchell [14], and so far only for rectangular pieces of paper.

The only other paper of which we are aware that proves the existence of continuous folding processes is [8]. This paper proves that every convex polygon can be folded into a uniaxial base via Lang's universal molecule [25] without gussets. Furthermore, unlike [14], no additional creases are introduced during the motion, and each crease face remains flat. This result can be used to animate the folding process.

3.4 Map Folding: Sequence of Simple Folds

In contrast to the complex origami folds arising from reaching folded states [8, 14], we can consider the less complex model of simple folds. A *simple fold* (or *book fold*) is a fold by ±180° along a single line. Examples are shown in Figure 4. This model is closely related to "pureland origami" introduced by Smith [33, 34].

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 4): 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., [27]. Arkin et al. [1] 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.

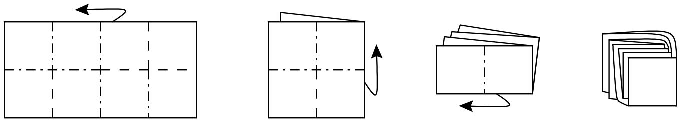

Figure 4: Folding a 2×4 map via a sequence of 3 simple folds.

Surprisingly, slight generalizations of map folding are (weakly) NP-complete [1]. Deciding whether a rectangle with horizontal, vertical, and diagonal (±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 becomes 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 a flat folded state.

4 Related Problems in Folding and Unfolding

Folding and unfolding is an area of geometry in which, like computational origami, there have been several new results in the past few years. In general, the problem of interest is how a particular object (e.g., a piece of paper, a linkage, or a polyhedron) can be folded subject to some natural constraints. For general surveys on folding and unfolding, see [7, 30].

4.1 Linkage Folding: One-Dimensional Origami

Consider a one-dimensional piece of paper, a line segment, marked at certain points with creases. It is most natural to fold such a piece of paper in the plane; a flat folding ends with the paper back on a line. Every one-dimensional piece of paper has a flat folded state, by alternating the creases between mountain and valley. If the mountain-valley assignment is prescribed, the crease pattern may not be flat-foldable, but this can be tested in linear time [1]. Indeed, it is proved in [1] that, in one dimension, the existence of a flat folded state is equivalent to foldability via a sequence of simple folds.

Now consider the problem of reaching a folded state, not necessarily flat, by a continuous motion. We think of the creases (points) as *hinges*, and the edges between the creases as rigid *bars*. Can we continuously fold the one-dimensional piece of paper into a desired folded state while remaining in the plane and without introducing new creases? This carpenter's rule problem is usually stated in the (equivalent) reverse direction: can every polygonal chain be straightened out without self-intersection and while preserving the bar lengths? Connelly, Demaine, and Rote [6] recently proved that the answer is always yes. In addition, every polygon (polygonal loop) can be folded into a convex configuration. Some examples of these motions are shown in Figure 5. Streinu [35] has developed another motion based on [6].

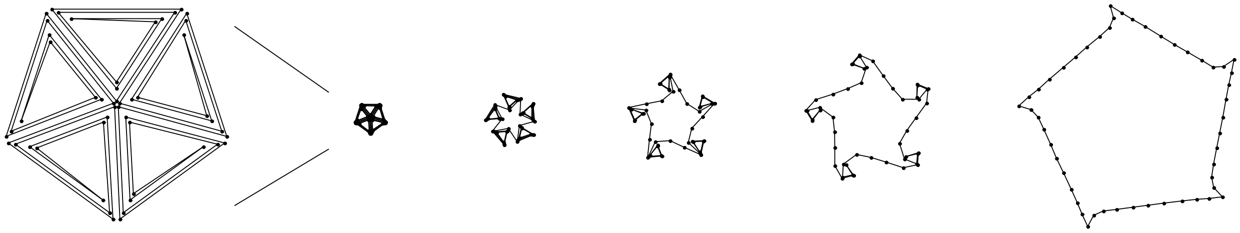

Figure 5: Convexifying the "doubled tree" on the left. See `http://daisy.uwaterloo.ca/~eddemain/linkage/` for more animations.

4.2 Folding and Unfolding 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. For example, can every convex polyhedron be cut along its edges and unfolded into one flat piece without overlap? This classic problem, dating back to 1525, remains open. However, if cuts are allowed interior to the faces of the polygon, there is a one-piece unfolding without overlap. See [7, 30] for details.

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. This problem has been studied by Lubiw and O'Rourke [26], and by Demaine, Demaine, Lubiw, and O'Rourke [12]. A particularly surprising discovery from this work [26] is that the well-known 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 6 for crease patterns and gluing instructions.

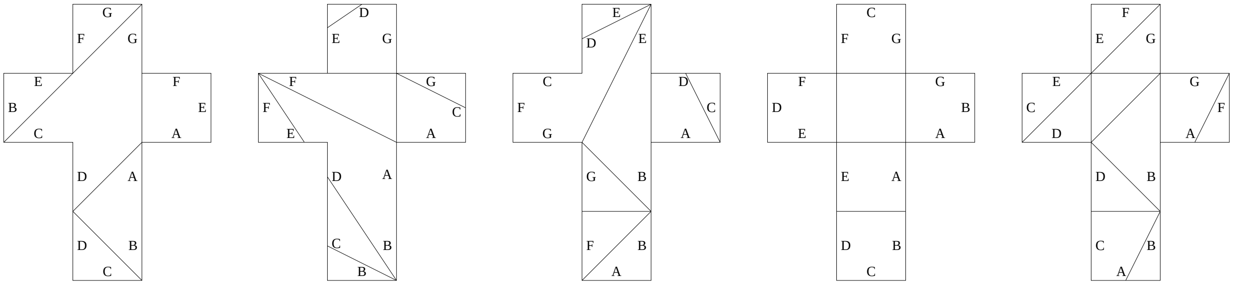

Figure 6: The five edge-to-edge gluings of the cross [26]. See `http://daisy.uwaterloo.ca/~eddemain/metamorphosis/` for a video animation of the foldings.

References

[1] 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? In *Proceedings of the 7th International Workshop on Algorithms and Data Structures*, Providence, Rhode Island, August 2001.

[2] Paulo Taborda Barreto. Lines meeting on a surface: The "Mars" paperfolding. In Koryo Miura, editor, *Proceedings of the 2nd International Meeting of Origami Science and Scientific Origami*, pages 323–331, Otsu, Japan, November–December 1994.

[3] Alex Bateman. Computational tools for origami tessellations. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, Monterey, California, March 2001.

[4] 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*, 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.

[5] 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. Reprinted in *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, 2001.

[6] Robert Connelly, Erik D. Demaine, and Günter Rote. Straightening polygonal arcs and convexifying polygonal cycles. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 432–442, Redondo Beach, California, November 2000.

[7] Erik D. Demaine. Folding and unfolding linkages, paper, and polyhedra. In *Proceedings of the Japan Conference on Discrete and Computational Geometry*, Lecture Notes in Computer Science, Tokyo, Japan, November 2000.

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

[9] 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.

[10] 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.

[11] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Flattening polyhedra. Manuscript, 2000.

[12] Erik D. Demaine, Martin L. Demaine, Anna Lubiw, and Joseph O'Rourke. Enumerating foldings and unfoldings between polygons and polytopes. In *Proceedings of the Japan Conference on Discrete and Computational Geometry*, Lecture Notes in Computer Science, Tokyo, Japan, November 2000.

[13] 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.

[14] Erik D. Demaine and Joseph S. B. Mitchell. Reaching folded states of a rectangular piece of paper. Manuscript, May 2001.

[15] S. Fujimoto. *Sojo suru origami asobi no shotai (Invitation to creative origami playing).* Asahi Culture Center, 1982.

[16] Martin Gardner. Paper cutting. In *New Mathematical Diversions (Revised Edition)*, Spectrum Series, chapter 5, pages 58–69. The Mathematical Association of America, Washington, D.C., 1995.

[17] Branko Grünbaum and G. C. Shephard. *Tilings and patterns.* W. H. Freeman, 1987.

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

[19] Thomas Hull. Counting mountain-valley assignments for flat folds. *Ars Combinatoria*, 2001. To appear.

[20] 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.

[21] 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.

[22] Toshikazu Kawasaki and Masaaki Yoshida. Crystallographic flat origamis. In H. Huzita, editor, *Proceedings of the 1st International Meeting of Origami Science and Technology*, pages 223–227, Ferrara, Italy, December 1989. An unabridged Japanese version appeared in *Sasebo College of Technology Report*, 24:101–109, 1987.

[23] 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.

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

[25] Robert J. Lang. Trees and circles: an efficient algorithm for origami design. In *Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, Monterey, California, March 2001.

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

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

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

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

[30] 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.

[31] Chris K. Palmer. *Bay Area Rapid Folders Newsletter.* Issues January 1994, March 1994, Winter 1995, Winter 1996, Spring 1996, Summer 1996, and Spring 1997. The last article is joint with Paulo Barreto.

[32] Chris K. Palmer. Extruding and tesselating polygons from a plane. In Koryo Miura, editor, *Proceedings of the 2nd International Meeting of Origami Science and Scientific Origami*, pages 323–331, Otsu, Japan, November–December 1994.

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

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

[35] 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.

[36] Helena Verrill. Origami tessellations. In R. Sarhangi, editor, *Conference Proceedings of Bridges: Mathematical Connections in Art, Music, and Science*, pages 55–68, 1998.

[37] David Wells. Hinged tessellations. In *The Penguin Dictionary of Curious and Interesting Geometry*, pages 101–103. Penguin Books, 1991.