On the Design of Physical Folded Structures

by

Jason S. Ku

B.S., Massachusetts Institute of Technology (2009)
S.M., Massachusetts Institute of Technology (2011)

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author Signature redacted ....
Department of Mechanical Engineering
May 18, 2016

Certified by.................... Signature redacted
Sanjay E. Sarma
Professor
Thesis Supervisor

Signature redacted
Accepted by ....
Rohan Abeyaratne
Chairman, Department Committee on Graduate Theses

On the Design of Physical Folded Structures

by

Jason S. Ku

Submitted to the Department of Mechanical Engineering
on May 18, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

Folding as a subject of mathematical, computational, and engineering study is relatively young. Most results in this field are hard to apply in engineering practice because the use of physical materials to construct folded structures has not been fully considered nor adequately addressed. I propose a three-fold approach to the design of folded structures with physical consideration, separating for independent investigation (1),the computational complexity of basic folding paradigms, (2) the automated accommodation of facet material volume, and (3) the design of folded geometry under boundary constraints. These three topics are each necessary to create folded structures from physical materials and are closely related.

Thesis Supervisor: Sanjay E. Sarma
Title: Professor

Acknowledgments

This thesis is dedicated to my wife and parents who have supported me so much throughout my time as a doctoral candidate. I also can't even begin to thank my thesis committee and all the other collaborators that have shared in my research journey. You inspire me everyday to learn more about the world and to teach others what we may find.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 2016, modern technology enables us to design and fabricate many beautiful and complex devices. It is the age of computer controlled machining: CNC machine tools, the 3D printer, the water jet, the laser cutter. These tools are built to fabricate a single part or assembly, and they do so magnificently. However, the future of fabrication lies not in the creation of better parts, but the design of single parts that can transform for many applications. The future is transformers.

Transformers have existed in the realm of science fiction for decades. Hollywood would have us believe that these transformers are made from big blocks of metal that can take the shape of your choice of vehicle and humanoid robot. But machines that change between two states is quite limited. We would prefer the liquid metal from Terminator, able to transform fluidly into any object, giving rise to the study of self-assembly: many simple particles that can assemble by themselves into complex shapes. However such a system has drawbacks, namely the system might be much too unconstrained. How would the particles interact or even stay together? Perhaps the set of particles could be locally constrained to each other in some way, forming a network of connections that could then transform locally to construct the global object. Perhaps a sheet of connections would suffice?

Imagine owning not a 3D printer, but a physical 3D display. It is not a flat display hung up on the wall that you look into. It is not a 3D hologram that gives the illusion of the presence of an object. Instead, imagine a programmable sheet of material that

Figure 1-1: Images of transformers in popular culture. The left is Optimus Prime from the Transformers series, and the right depicts a T-1000 from the Terminator series.

can take the shape of any form that you wish to imagine. Such a device would be the pinnacle of rapid prototyping. You wouldn't need to wait for a machine to cut through material, or a tool head to traverse a voxel grid. The sheet simply transforms in parallel, assembling in seconds into the object you desire. Fully reshapeable, fully reusable: the ultimate modular design. How does one accomplish such a grand vision? The answer is folding.

1.1 Scope

This thesis presents multiple results in the field that is now often referred to as *computational origami*, in the direction of computing and designing transformable structures using folding.

Chapter 2 analyzes the computational complexity of many folding related paradigms that are applicable to engineering design. In particular, multiple decision problems related to both *flat folding*, creating folded states that lie in a plane, and *simple folding*, creating folded states that may only be folded along a single line at a time, are shown to be NP-complete, improving and correcting major results in the field.

Chapter 3 introduces the *offset crease method*, a new technique compensating for material volume in flat-foldable crease patterns. This method exchanges an increase in degrees of freedom to allow for facets to be separated in the folded state, while allowing for a full range of motion to the "flat" state. A software implementation of the algorithm, and physical models were produced to demonstrate the technique.

Lastly, Chapter 4 presents a new paradigm for designing crease patterns from prescribed boundary conditions. This technique is quite broad in its scope, generalizing different formulations of Fold-and-Cut, the Universal Molecule, and even tesselation generation. The main result is a necessary and sufficient condition for an isometry to exist for a given boundary condition, and a procedure for producing any isometry. For a very specific model, we also explore crease patterns that can fold to multiple boundary conditions, and use the result to generate terrains.

1.2 Background

People have folded sheets of material to create new forms for thousands of years, an activity most notably associated with the ancient paper folding tradition known as *origami* (折紙 in Japanese). While the word origami refers specifically to folding paper (紙), *folding* as a transformational procedure can naturally be applied to any sheet-like surface. For most of its history, origami has comprised a static repertoire containing a handful of traditional forms. However in recent years, modern origami has exploded into an expressive art form, in large part due to the growth of folding into its own rich mathematical and computational field, with mechanical engineering exploring its applications. Already, the study of folding mechanisms has influenced a diverse set of disciplines, as summarized in the following paragraphs.

Folding has long been employed in **manufacturing**, with applications in packaging and sheet metal bending. Current packaging research investigates optimal layout for die-cut unfoldings [40] as well as path planning for how to fold complex wrappings without self-intersection [61]. Similar concerns are studied with respect to sheet metal [60] in addition to work modeling and accounting for the elasto-mechanical properties of folding metal [24]. While there is exciting current work in building multi-armed robots to bend sheet metal into complex three-dimensional shapes [56], most sheet metal folding still uses traditional straight-line mechanical brakes. This type of folding is called *simple-folding* because facets are restricted to fold around a single straight line at a time, which are much easier to produce than more complex folds. Robots have been developed specifically to perform simple folds [6].

Recently the fields of **MEMS** and **robotics** have both used folding as a tool to build self-assembling three-dimensional structures, where traditional fabrication and assembly techniques are difficult to apply. Folding has been used to create a variety of MEMS components such as electrostatically actuated mirrors [26], capacitors [35], and meta-materials [5]. A feature of folding three-dimensional structures on the micro-scale is the possibility of forming a high density of structural and electrical connections which are not possible with relatively planar fabrication techniques, allowing the construction of intricate circuitry [31] for advanced computing hardware. After patterning electronics on a foldable surface, micro-robots can self-assemble along predefined fold lines [25], while actuating embedded folding linkages [27]. Such folding motions are an active area of research in motion-planning [50]. Printable robotics has emerged as its own field in the creation of affordable, even disposable micro-robots [45]. On the other hand, programmable surfaces dream of creating a single generic material that can reconfigure and update its own hardware [28], similar to software updates today.

Folding also has application in **biology** and the design of **biomedical** devices. Biological structures such as proteins [43] and DNA [47][21] each fold as a one-dimensional chain with three-dimensional connections to itself. While the study of folding is certainly closely tied to these structures, the work proposed here will focus on building three-dimensional structures from two-dimensional surfaces, not one-dimensional chains. Such structures are already being investigated to design expandable stents [37], medical implants conforming to specific geometry [41], and scaffolds upon which replacement organs might be grown [34].

The versatility and aesthetic nature of folded forms also lends itself to the field of **design**, especially as applied to **architecture** and deployable structures in transportation and **space**. Folding is being used to design transformable spaces [14] and dynamic, parametric buildings [42] that can adapt according to style or function. Folding has immediate military application when designing deployable shelters [57] and constructing temporary adaptable bridges [3]. The design of automotive airbags has been aided by modeling collapsed states as folded structures [13]. Sending machines to outer space requires very constrained design. Equipment must fit into a relatively small rocket before either being assembled or deployed in space. Because space walks and robotic assembly is dangerous and expensive, automatic deployment using folding mechanisms has become ubiquitous in the design of new large aperture telescopes [22] and solar panel arrays [49].

This thesis focuses on folding two-dimensional surfaces. Folding one-dimensional chains (linkage folding) has a long history in engineering and manufacturing; on the other hand, folding two-dimensional surfaces has only recently begun being studied quantitatively, the added new dimension of constraints and complexity creating a modern field ripe for exploration. Young fields like surface folding are where small advances can make a big impact on a wide variety of applications like those described above. Each application requires the design of folded structures that satisfy three main requirements:

- **[Intersection]** foldings do not cause collision of material with itself or its surroundings.
- **[Material]** foldings can accommodate and be fabricated using available materials;
- **[Geometry]** foldings adhere to some starting sheet and final shape criteria;

While it is tempting to try and tackle all three design requirements at once within a single framework, the complexity of conforming to each individually suggests separating them into modular components which may be solved, and then ultimately reassembled to solve a particular problem. Of course, all three requirements are necessary for the construction of real, physical folded structures, though for different applications, some requirements may more important than others. For example, the design of a deployable solar sail or orbital telescope may have very specific geometric and material requirements in order to function, but may allow folding panels to mechanically cross, provided the nature of the interactions can be precomputed. Alternatively, an implantable medical device for releasing chemicals at a constant rate may require a non-crossing watertight seal out of biocompatible materials, but the geometry may be allowed to take many forms. Origami design itself is concerned with producing precise geometry without cutting the material, though often the material properties of paper may safely be ignored. One would like to develop general tools to address each requirement in turn in order to tailor a design to a specific application.

This section analyzes each requirement in isolation, exploring the existing approaches, solutions, and difficulties involved with each. Then, novel research paths toward addressing each requirement will be proposed. The narrative behind this proposal is not structured around tackling a single application, but instead hopes to develop general algorithms that may be combined together to design and fabricate complex and realizable physical folded structures.

1.2.1 Intersection

The main reason that folding is such a difficult problem is that different parts of complicated two-dimensional folding linkages may collide, and checking for such non-local intersections can be computationally intractable. In addition to the geometry of folded structures (the location of each vertex and edge in space), facets in a folding that touch also must be prescribed a *layer ordering* in order to fully define the topology of the folding. Little is known about efficiently constructing folded states with valid layer orderings where intersections between facets are forbidden.

If we take a flat-folded structure and unfold it, the creases left behind form a *flat-foldable crease pattern* (a *crease pattern* referring to a straight-line embedding of a planar graph). It is known that checking each part of the crease pattern for local intersections is easy to compute [33]. However, given a crease pattern, optionally *assigned* with each crease labeled either mountain (the paper folds backwards) or valley (the paper folds forwards), the *flat-foldability problem* asks whether the crease pattern comes from some flat folding. This decision problem was shown to be NP-complete for both assigned and unassigned crease patterns [9]. However, it is unknown if the flat-foldability problem is solvable in polynomial time for even very simple classes crease patterns. We show that this problem is still NP-complete for *box-pleated crease patterns* (angles between creases restricted to multiples of 45°, with application to programmable grids for transformable robotics [28]).

Of course finding a motion folding a flat surface to a non-intersecting folded state that avoids self-intersection along the way is an even more difficult problem. While the study of general non-intersecting rigid motions of folded structures is an attractive and notoriously difficult topic of current research [51], a more restricted problem studies a subset of flat-folding with direct application to manufacturing: that is *simple-foldability*, deciding whether a two-dimensional crease pattern can be folded by a sequence of simple folds without self-intersection. Informally, a simple fold can only rotate paper around a single rotation axis before returning the paper back to the plane. Arkin et al. [4] introduced many models of simple-folds, proving that deciding simple-foldability is weakly NP-complete for some of them, and that simple-foldability can be solved in polynomial time for rectangular paper with paper-aligned orthogonal creases. We prove these models strongly NP-complete, in addition to proving hardness for additional models. However, in order to fabricate flat-folded structures from physical materials, we must consider the possibility that the paper might have non-negligible thickness.

1.2.2 Material

While the design of specific geometry may produce outputs that satisfy provided geometric criteria, there can be problems when applying these algorithms when their outputs assume a structure folded from zero-thickness, zero-volume materials. In engineering, structures must be built using physical materials where the volume of the surface cannot safely be ignored. For example, when designing a complex electric circuit with many layers of components folded on top of one another, the components and the substrate on which they reside have thickness that must be considered and aligned. At a larger scale, architectural and astronautical folded structures made of thick structural materials must be handled. There are many existing approaches to account for material thickness when designing folding structures, each with their own strengths and weaknesses. These techniques are discussed in Chapter 3.

We present a new method for accounting for material volume that widens creases in a systematic way without relying on flexible materials. While such a technique might not preserve the exact structure of the input crease pattern, it could create a structure that might be easier to fabricate than other techniques described above. Facet surfaces in the produced structure's unfolded state could be made coplanar, allowing for straightforward fabrication in a layer-by-layer manufacturing processes. These same surfaces could be made parallel in the produced structure's folded state, allowing any surface mounted components to mate naturally.

The approach converts flat-foldings into facet-separated foldings by replacing each flat crease in the input crease pattern by two parallel creases symmetrically offset about the original at a distance proportional to an assigned crease width satisfying certain properties of the original crease pattern. Instead of one crease folding flat with a turn angle of 180°, the two new creases would have final turn angles of 90°. This crease widening creates difficulties at crease pattern intersections since the offset creases would no longer converge to a point. Material in the vicinity around each crease pattern vertex is discarded to accommodate crease widening. While such a modification creates holes in the material, the process introduces new degrees of freedom, allowing the widened creases to fold.

Because of their reliance on perfectly thin materials, most (if not all) of the algorithms developed to design folded structures have not been useful in designing structural mechanisms out of physical materials, likely because the existing techniques discussed above restrict range of motion and are difficult to fabricate in practice. The introduction of this new method that accommodates material thickness while addressing these two issues can open the door to the application of existing geometric design algorithms, allowing for the practical engineering design of physical folded structures. However, it is not sufficient to simply accommodate material thickness. One must also be able to generate a folding that adheres to given design parameters.

1.2.3 Geometry

This section explores existing algorithmic design techniques for creating folded structures satisfying certain requirements. Not surprisingly, there is no single approach that is appropriate for every application, and each available method for producing folded structures conforms to very specific inputs and outputs, while new algorithms will likely be similarly constrained. It is interesting to note that each existing approach presented in this section deterministically provides a unique output for a given valid input, optimizing over some tightly constrained space. While some may interpret this property as a desirable, engineering and design often are not so tightly constrained. It could be more useful to find a more general framework for producing families of folded structures that could then be adjusted or optimized based on the requirements of a specific application.

Discussed in this section are three folding design algorithms that are subtly related. The first two, tree theory and fold-and-cut both design *flat-foldings*, structures that lie in a single flat plane in their folded state. The third is Origamizer, an algorithm for producing three-dimensional polyhedral surfaces. Each of these methods share a common thread: each requiring a partition of the surface to be folded into regions, and then filling these regions with creases so that the boundary of each region folds in a pre-specified manner. To be consistent with the literature, I use the word "paper" to refer to the planar surfaces that are to be folded, keeping in mind that the design algorithms are readily applicable to other materials.

Tree theory is a method proposed by Lang for designing flat-foldable structures that each represent an underlying stick figure [38][39]. Given a sheet of paper and a weighted tree, the algorithm constructs a *uniaxial base*, informally a folding of the paper corresponding to the tree comprising independent flaps connected in the same topology and proportion as the tree. The algorithm works by modeling each edge of the input tree by a strip of paper with constant width, and arranging them on the paper without overlap using a non-convex optimization maximizing scale. This optimization partitions the paper into smaller convex regions called *molecules.* Then creases are filled into each molecule using the *universal molecule*, a unique set of creases folding the boundary of each molecule onto a single line consistent with the input tree as desired [10]. The universal molecule is constructed by insetting every side of the paper at constant speed, with the output creases formed by the trace of polygon vertices during the sweep. While it is conjectured that this algorithm always provides folded structures that do not self-intersect, such questions are notoriously difficult to prove. Thus we discuss self-intersection detection separately.

The **fold-and-cut** problem questions how to fold a piece of paper upon which a polygon (or more generally some straight-line graph) is drawn so that a single straight cut can be made, cutting precisely the lines drawn and nothing more. Two methods are known for solving this problem, but only one is applicable generally. The first method proposed by Demaine et. al. creases the paper along the *straight skeleton* of the graph drawn, along with some additional perpendicular creases [16]. This method is easy to fold, but is not applicable generally; an infinite number of creases must be introduced in order to fold some instances. The second method proposed by Bern et. al. instead decomposes the faces of the graph drawn into triangles and quadrilaterals with nice properties, and then orients and manipulates the flaps to align only the drawn line segments together [8]. Impressively, they were able to show that there always exists a layer ordering for output flat-foldings that avoids self-intersection of the paper [20]. In both algorithms, solutions found are uniquely determined, and both require the alignment of polygonal boundaries into specified geometry.

The third design algorithm is used to construct general three-dimensional surfaces. **Origamizer**, designed by Tachi takes as input a triangulated connected polyhedral surface, lays out the individual triangles onto a convex paper by solving an optimization problem, and then fills in creases on the paper between the triangles that brings the triangles together [53]. Again, it is conjectured that the produced folded structure can be arranged in a way that avoids self-intersection, but the claim has not yet been proven. However, the algorithm has been used successfully to construct very complex surfaces. Similar to tree theory, this algorithm has two stages: a layout/optimization stage, and crease generation stage. When generating creases, the paper between the surface triangles must fold so that the boundaries of the inputs facets are brought into alignment with each other.

A common thread running through each of these design algorithms is the problem of mapping the boundary of the paper to some specified folded configuration. In tree theory, the boundary must be mapped onto a doubly-covered stick figure, in the fold-and-cut problem the drawn segments must be made collinear, while Origamizer must fold triangle boundaries onto the wireframe of the input model. Each of these subproblems then can be thought of as a special case of a more general problem called the **hole problem**: given a sheet of paper and a prescribed folding of its boundary, is there a way to fold the paper's interior without stretching so that the boundary lines up with the prescribed boundary condition?

A solution to the hole problem might be used to solve many of the above results in a novel way, as well as address some new applications. For example, when trying to combine separately designed parts of a folded structure, a solution to the hole problem could be used to design an interface between them. We present a necessary and sufficient condition for a solution to exist, and give a procedure for generating any isometry consistent with the boundary condition. The idea is to approach crease filling in a manner similar to the construction of the universal molecule, but instead of insetting the entire perimeter in tandem, only insetting a single vertex at a time to break up the polygon into smaller and smaller subproblems. A feature of this approach is that it is able to produce many solutions for creasing the paper to satisfy a provided boundary condition. Instead of a unique solution provided by existing algorithms, a family of solutions can be explored tailoring to the specifications of the application. One could imagine that among a vast family of solutions, finding one that avoids self-intersection might be tractable, though we have not yet been able to find any nontrivial necessary or sufficient conditions to avoid self-intersection.

Our approach applies to both flat-foldings as well as more general three-dimensional structures.

Chapter 2

Complexity

This chapter relates to results that have been previously published. Section 2.1 represents joint work with Akitaya et al. [1] accepted for publication in the conference proceedings of JCDCGG 2015. Section 2.2 represents joint work with Akitaya et al. [2]. This work has been presented at multiple venues, but the detailed proofs provided here have not yet been published.

2.1 Box-Pleating is Hard

In their seminal 1996 paper, Bern and Hayes initiated investigation into the computational complexity of origami [9]. They claimed that it is NP-hard to determine whether a given general crease pattern can be folded flat, both when the creases have or have not been assigned crease directions (mountain fold or valley fold). Since that time, there has been considerable work in analyzing the computational complexity of other origami related problems. For example, Arkin et al. [4] proved that deciding foldability is hard even for simple folds, while Demaine et al. [18] proved that optimal circle packing for origami design is also hard.

While the gadgets in the hardness proof presented in [9] for unassigned crease patterns are relatively straightforward, their gadgets for assigned crease patterns are considerably more convoluted, and quite difficult to check. In fact, we have found an error in even their unassigned crossover gadget where signals are not guaranteed to transmit correctly for wires that do not cross orthogonally, which is required in their construction. Part of the reason no one found this error until now is that there was no formal framework in which to prove statements about flat-folded states. We attempt to provide such a framework.

At the end of their paper, Bern and Hayes pose some interesting open questions to further their work. While most of them have been investigated since, two in particular (problems 2 and 3) have remained untouched until now. First, is there a simpler way to achieve a proof for assigned crease patterns (i.e. "without tabs")? Second, their reductions construct creases at a variety of unconstrained angles. Is deciding flat foldability easy under more restrictive inputs? For example, *box pleating* involves folding creases only along on a subset of a square grid and the diagonals of the squares, a special case of particular interest in transformational robotics and self-assembly, with a universality result constructing arbitrary polycubes using box pleating [7].

In this section we address both these questions. We prove that deciding flat foldability of box-pleated crease patterns is NP-hard in both the unassigned and assigned cases, using relatively simple gadgets containing no more than 25 layers at any point.

2.1.1 Definitions

In general, we are guided by the terminology laid out in [20] and [46]. An *isometric flat folding* of a paper P is a function $f : P \to \mathbb{R}^2$ such that if γ is a piecewise-geodesic curve on P parameterized with respect to arc-length, then $f(\gamma)$ is also a piecewise-geodesic curve parameterized with respect to arc-length. It is not hard to show that under these conditions f must be continuous and non-expansive. Let X_f be the boundary of a paper P together with the set of points not differentiable under f. Then one can prove that X_f is a straight-line graph embedded in the paper [46], with vertex set V_f and edge set C_f, the *creases* of our folding f. A vertex or crease in V_f or C_f is *external* if it contains a boundary point of P, and *internal* otherwise. Subtracting X_f from P results in a disconnected set of open polygons F_f we call *faces.* For any face $F \in F_f$, $f(F)$ is either an isotopic transformation in $\mathbb{R}^2$, or the transformation involves a reflection and is anisotopic. Define $u_f : P \setminus X_f \to \{-1, 1\}$ such that $u_f(p) = -1$ if the face containing p is reflected under f and $u_f(p) = 1$ otherwise. We call $u_f(p)$ the *orientation* of the face containing p. Every point in P is in exactly one of V_f, C_f, or F_f. We call this partition of P the *isometrically flat foldable crease pattern* $\Sigma_f = (V_f, C_f, F_f)$ induced by f. We call a folding *box pleating* if every vertex lies on two dimensional integer lattice, and the creases are aligned at multiples of 45° to each other.

We say two disjoint simply connected subsets of P are *adjacent* to each other if their closures intersect; we call such an intersection the *adjacency* of the adjacent subsets. We say a simply connected subset of P is *uncreased* under f if f is injective when restricted to the subset. We say two simply connected subsets of P *overlap* under f if the interiors of their images under f intersect. We say two simply connected subsets of P *strictly overlap* under f if their images under f exactly coincide. It is known that the set of creases adjacent to an internal vertex of a crease pattern obey the so called Kawasaki-Justin Theorem: the alternating sum of angles between consecutive creases when cyclically ordered around the vertex equals zero [20]. This condition turns out to be necessary sufficient: given a paper P exhaustively partitioned into a set of isolated points V, open line segments C, and open disks F such that every point in V is adjacent to more than two segments in C, then (V, C, F) is an isometrically flat foldable crease pattern induced by a unique isometric flat folding if and only if (V, C, F) obeys the Kawasaki-Justin Theorem.

Let a function $\lambda_f : P \times P \to \{-1, 1\}$ be a *global layer ordering* of an isometric flat folding f if it obeys the following six properties.

Existence: λ_f satisfies *existence* if $\lambda_f(p, q)$ is defined for every distinct pair of points p and q that strictly overlap under f and at least one of p or q is not in X_f; otherwise $\lambda_f(p, q)$ is undefined. Informally, order is only defined between a point on a face and another point overlapping it in the folding.

Antisymmetry: λ_f is *antisymmetric* if $\lambda_f(p, q) = -\lambda_f(q, p)$, where λ_f is defined. Informally, if p is above q, then q is below p.

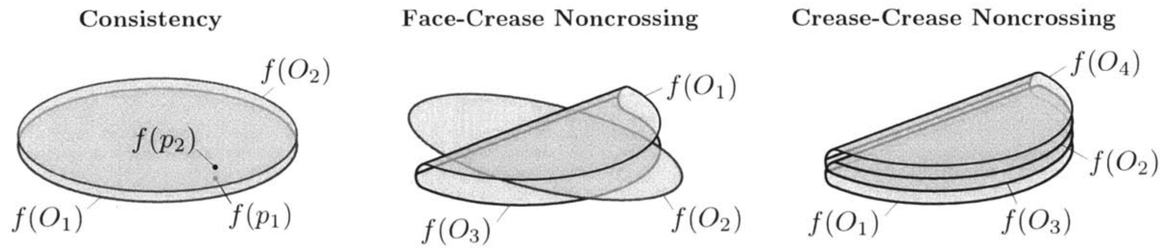


Figure 2-1: Topologically different local interactions within an isometric flat folding. Forbidden configurations are shown for Face-Crease and Crease-Crease Non-Crossing.

Transitivity: λ_f is *transitive* if $\lambda_f(p,q) = \lambda_f(q,r)$ implies $\lambda_f(p,r) = \lambda_f(p,q)$, where λ_f is defined. Informally, if q is above p and r is above q, then r is above p.

Consistency (Tortilla-Tortilla Property): For any two uncreased simply connected subsets O_1 and O_2 of P that strictly overlap under f, λ_f is *consistent* if $\lambda_f(p_1, p_2)$ has the same value for all $(p_1, p_2) \in O_1 \times O_2$, where λ_f is defined. See Figure 2-1. Informally, if two regions completely overlap in the folding, one must be entirely above the other.

Face-Crease Non-crossing (Taco-Tortilla Property): For any three uncreased simply connected subsets O_1, O_2, and O_3 of P such that O_1 and O_3 are adjacent and strictly overlap, and O_2 overlaps the adjacency between O_1 and O_3 under f, λ_f is *face-crease non-crossing* if $\lambda_f(p_1, p_2) = -\lambda_f(p_2, p_3)$ for any points $(p_1, p_2, p_3) \in O_1 \times O_2 \times O_3$, where λ_f is defined. See Figure 2-1. Informally, if a region overlaps a nonadjacent internal crease, the region cannot be between the regions adjacent to the crease.

Crease-Crease Non-crossing (Taco-Taco Property): For any two adjacent pairs of uncreased simply connected subsets (O_1, O_2) and (O_3, O_4) of P such that every pair of subsets strictly overlap and the adjacency of O_1 and O_2 strictly overlaps the adjacency of O_3 and O_4 under f, λ_f is *crease-crease non-crossing* if either $\{\lambda_f(p_1, p_3), \lambda_f(p_1, p_4), \lambda_f(p_2, p_3), \lambda_f(p_2, p_4)\}$ are all the same or half are $+1$ and half are -1, for any points $(p_1, p_2, p_3, p_4) \in O_1 \times O_2 \times O_3 \times O_4$, where λ_f is defined. See Figure 2-2. Informally, if two creases overlap in the folding, either the regions incident to one crease lie entirely above the regions incident to the other (all same), or the regions incident to one crease nest inside the regions incident to the other (half-half).

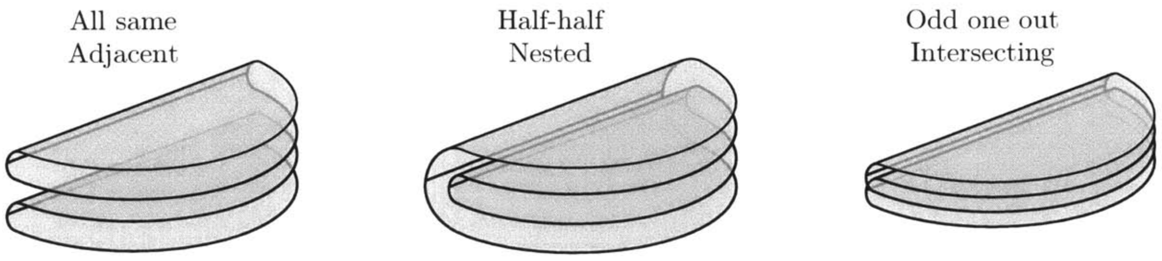


Figure 2-2: Local interaction between overlapping regions around two distinct creases.

If there exists a global layer ordering for a given isometrically flat foldable crease pattern, we say the crease pattern is *globally flat foldable*. Consider an isometrically flat foldable crease pattern Σ_f containing two adjacent uncreased simply connected subsets O_1 and O_2 of P that strictly overlap under f, and let p and q be points in O_1 and O_2 respectively that overlap under f. O_1 and O_2 are subsets of disjoint adjacent faces of the crease pattern mutually bounding a crease. If λ_f is a global flat folding of Σ_f, then it induces a *mountain/valley assignment* $\alpha_{\lambda_f}(c) = u(p)\lambda_f(p,q)$ for each crease point c in the adjacency of O_1 and O_2. This assignment is unique by consistency. We call a crease point c a *valley fold* (V) if $\alpha_{\lambda_f}(c) = 1$ and a *mountain fold* (M) if $\alpha_{\lambda_f}(c) = -1$. In the figures, mountain folds are drawn in red while valley folds are drawn in blue. By convention, if $\lambda_f(p,q) = -1$ we say that p is *above* q, and if $\lambda_f(p,q) = 1$ we say that p is *below* q.

Given an isometrically flat foldable crease pattern Σ_f, the UNASSIGNED-FLAT-FOLDABILITY problem asks whether there exists a global layer ordering for f. Alternatively, given an isometrically flat foldable crease pattern Σ_f and an assignment $\alpha : C_f \to \{M, V\}$ mapping creases to either mountain or valley, the ASSIGNED-FLAT-FOLDABILITY problem asks whether there exists a global layer ordering for f whose induced mountain valley assignment is consistent with α.

We now prove the following implied properties of globally flat foldable crease patterns relating the layer order between points contained in multiple overlapping faces. Informally, *Pleat-Consistency* says if a face is adjacent and overlapping two larger faces, then the creases between them must have different M/V assignment, forming a pleat. *Path-Consistency* says that a face overlapping creases connecting an adjacent sequence of faces is either above or below all of them.

Lemma 2.1.1. (Pleat-Consistency) *If Σ_f is a globally flat foldable crease pattern containing disjoint uncreased simply connected subsets O_1, O_2, and O_3 of P with O_2 adjacent to both O_1 and O_3 such that O_2 strictly overlaps subsets $O'_1 \subset O_1$ and $O'_3 \subset O_3$, and the interiors of O_1 and O_3 overlap the adjacencies of O_2, O_3 and O_1, O_2 respectively, then $\lambda_f(p_1, p_2) = \lambda_f(p_2, p_3)$ for any pairwise overlapping points $(p_1, p_2, p_3) \in O_1 \times O_2 \times O_3$.*

Proof. Taco-Tortilla applied to O_3 which overlaps the adjacency of strictly overlapping sets O_2 and O'_1 implies $\lambda_f(p_2, p_3) = -\lambda_f(p_3, p_1)$. Similarly, Taco-Tortilla applied to O_1 which overlaps the adjacency of strictly overlapping sets O'_3 and O_2 implies $\lambda_f(p_3, p_1) = -\lambda_f(p_1, p_2)$, so $\lambda_f(p_1, p_2) = \lambda_f(p_2, p_3)$. □

Lemma 2.1.2. (Path-Consistency) *If Σ_f is a globally flat foldable crease pattern containing uncreased simply connected subset T of P and a disjoint sequence of adjacent uncreased simply connected subsets $O_1, \dots, O_n$ of P such that O_i strictly overlaps some subset T_i of T and the interior of O overlaps the adjacency of each pair O_i and O_{i+1} for $i = \{1, \dots, n-1\}$, then $\lambda_f(t_j, p_j) = \lambda_f(t_k, p_k)$ for any two pairs of overlapping points $(t_j, p_j) \in T_j \times O_j$ and $(t_k, p_k) \in T_k \times O_k$ for $j, k \in \{1, \dots, n\}$.*

Proof. If some O_i and O_{i+1} overlap, Taco-Tortilla and Consistency ensure that $\lambda_f(t_i, p_i) = \lambda_f(t_{i+1}, p_{i+1})$ for $(t_i, p_i) \in T_i \times O_i$ and $(t_{i+1}, p_{i+1}) \in T_{i+1} \times O_{i+1}$. Alternatively, O_i and O_{i+1} do not overlap and the closure of $O_i \cup O_{i+1}$ is an uncreased region for which $\lambda_f(t_i, p_i) = \lambda_f(t_{i+1}, p_{i+1})$ by consistency. Applying sequentially to each pair of faces proves the claim. □

The proofs in Section 2.1.4 and 2.1.5 contain many examples of the application of these properties. When proving the existence of a global layer ordering λ_f, it is often impractical to define λ_f between every pair of points. Frequently λ_f is uniquely induced by a M/V assignment, consistency, and transitivity. When it is not, we will provide λ_f between additional point pairs so that it will be. We present crease patterns with this implicit layer ordering information and encourage readers to fold them to reconstruct the unique layer orderings they induce.

2.1.2 Bern and Hayes and k-Layer-Flat-Foldability

Two crossover gadgets are presented in the reduction to UNASSIGNED-FLAT-FOLDABILITY provided in [9]. For each, they claim that the M/V assignment of the crease pair intersecting one edge of the gadget deterministically implies the M/V assignment of the crease pair on the opposite side. This claim is true for their perpendicular crossover gadget, but is unfortunately not true for the other for wires meeting at 45°. The gadget as described requires an exterior 45° angle between incoming wires that is the smallest angle at a four-crease vertex, forbidding the wires to be independently assigned by Pleat-Consistency. For completeness, we have also checked the family of possible gadgets of this form, with a rotated internal parallelogram, and no choice of rotation allows the gadget to function correctly as a crossover. Our proof to follow only uses the perpendicular crossover, avoiding this complication.

Also in [9], they define k-LAYER-FLAT-FOLDABILITY to be the same as UNASSIGNED-FLAT-FOLDABILITY or ASSIGNED-FLAT-FOLDABILITY but with the additional constraint that f maps at most k distinct points to the same point. They claim that their reduction implies hardness of UNASSIGNED-k-LAYER-FLAT-FOLDABILITY for $k = 7$. But in fact their perpendicular crossover gadget requires nine points to be mapped to the same point. Our reduction uses the same gadget as a crossover, so we reconfirm that UNASSIGNED-k-LAYER-FLAT-FOLDABILITY is NP-complete for $k \geq 9$, even for box pleated crease patterns. Also, because of the complexity of their assigned crease pattern reduction, they were unable to bound the number of layers in their reduction. We explicitly provide gadgets for the assigned case to prove ASSIGNED-k-LAYER-FLAT-FOLDABILITY is NP-complete for $k \geq 25$, even for box pleated crease patterns.

2.1.3 SCN-Satisfiability

Our reductions will be from the following NP-complete problem [48].

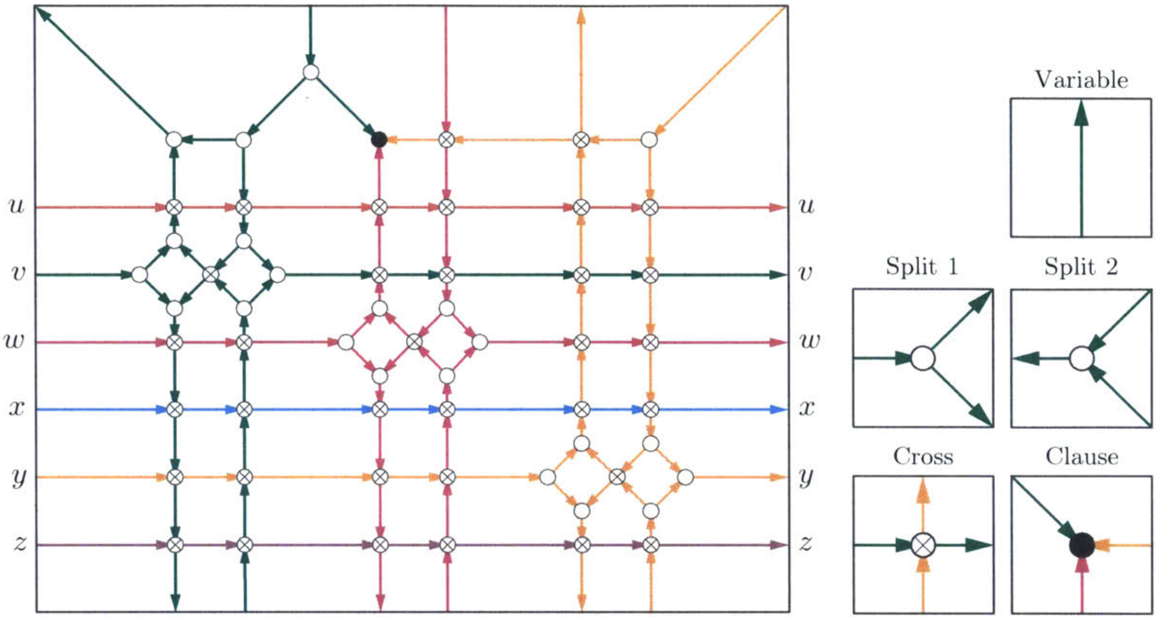


Figure 2-3: SCN Gadgets. [Left] A Complex Clause Gadget constructed from the Not-All-Equal clause on variables v, w, and y of a NAE3-SAT instance on six variables. [Right] The five elemental SCN Gadgets.

Problem 2.1.1. (Not-All-Equal 3-SAT) *Given a collection of clauses each containing three variables,* Not-All-Equal 3-SAT (NAE3-SAT)[1] *asks if variables can be assigned True or False so that no clause contains variables of only one assignment.*

We can construct a planar directed graph G embedded in $\mathbb{R}^2$ from an instance $\mathcal{N}$ of NAE3-SAT. For each clause, construct a Complex Clause Gadget as the one shown in Figure 2-3. The motivation behind the Complex Clause Gadget is to encode the bipartite graph implicit in $\mathcal{N}$ in a planar grid embedding that can be modularly connected. Each directed edge of the Complex Clause Gadget is associated with a different variable, and we associate a different color with each variable. Some variables do not participate in the clause and simply form a straight chain of directed segments from left to right. However, the three variables participating in the clause are rerouted to intersect at the black dot. We construct a Complex Clause Gadget for each clause in the instance of NAE3-SAT and chain them together side by side, so the arrows exiting the right side of one enter the left side of another. Graph G has vertices that

[1] *This problem is sometimes called 'positive' as variables cannot appear negated within clauses, however we follow the naming convention from [48].*

are adjacent to edges associated with exactly one, two, or three variables. We call these vertices *split*, *cross*, and *clause* vertices respectively. In the figures, they are labeled with white circles, crossed circles, and black circles respectively. We call such a directed graph G a *Split-Cross-Not-All-Equal* (SCN) graph.

Problem 2.1.2. (SCN-Satisfiability) *Given a SCN graph,* SCN-SATISFIABILITY *asks if variables can be assigned True or False so that no clause vertex is adjacent to edges associated with variables of only one assignment.*

The authors introduce SCN-Satisfiability as a useful intermediate problem because it is equivalent to NAE3-SAT but its embedding is planar, lies on a grid, and is constructed only by a small number of local elements. SCN-SATISFIABILITY is equivalent to NAE3-SAT because the bipartite graph connecting SCN variables to clause vertices is exactly the bipartite graph representing $\mathcal{N}$ by construction. However, G has useful structure for many problems. It is planar, the embedding contains edges with only four slopes, and the edges are directed meaning that a variable can be represented locally with respect to that direction. Further G is constructed from only a small number of local elements: a variable gadget, two split gadgets, a cross gadget, and a clause (simple) gadget as shown in Figure 2-3. We call these the five *elemental* SCN Gadgets. If we can simulate each of these gadgets in another context, proving that edges of the same color in each gadget must all have the same value, and edges adjacent to a clause vertex do not all have equal value, we can prove other problems NP-hard. This will be our strategy in the following sections.

Theorem 2.1.1. *If a problem X can simulate the elemental SCN gadgets such that edges of the same color in each gadget have the same value and edges adjacent to a clause vertex do not all have equal value and if the correspondent gadgets in X can be connected consistently, then X is NP-Hard.*

2.1.4 Unassigned Crease Patterns

In this section we present gadgets simulating the elemental SCN gadgets with unassigned crease patterns. They are shown in Figure 2-4.

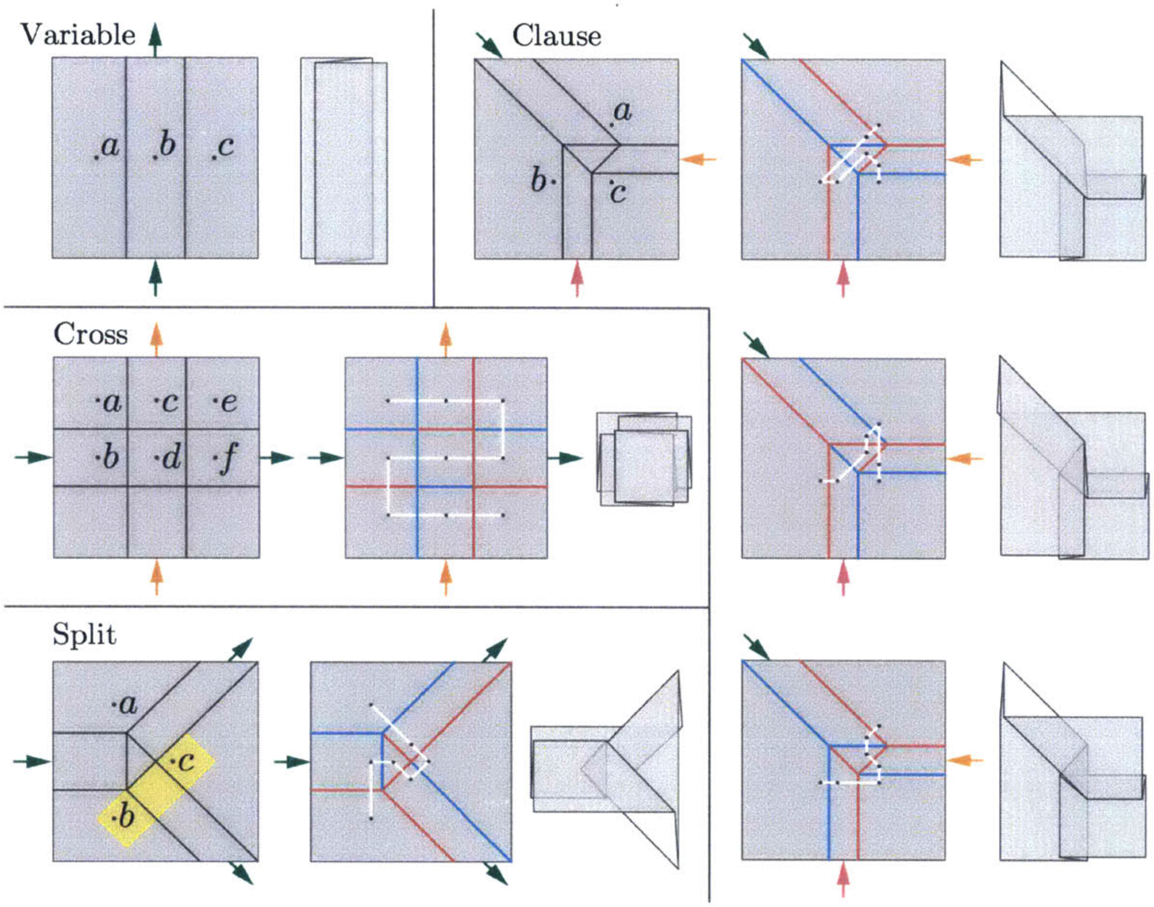


Figure 2-4: Elemental SCN Gadgets simulated with unassigned crease patterns.

We define a variable gadget to be a pair of parallel creases placed close together having an direction as shown in Figure 2-4. By pleat-consistency and transitivity, $\lambda_f(a,b) = \lambda_f(b,c) = \lambda_f(a,c)$ so, local to the gadget, it has exactly two globally flat foldable states. We say the variable is *True* if the face to the right of the variable direction is above the face to left ($\lambda_f(a,c) = 1$), and *False* otherwise.

Lemma 2.1.3. *The unassigned crossover gadget is a globally flat foldable crease pattern if and only if opposite variables are equal.*

Proof. Refer to Figure 2-4. Assume global flat foldability. Let A, B, C, D, E, F be the maximal subsets of the faces respectively containing points a, b, c, d, e, f such that every pair strictly overlap. First assume that $\lambda_f(a,b) = \lambda_f(c,d)$. By Taco-Taco with respect to adjacencies A, C and B, D, $\lambda_f(a,d) = \lambda_f(c,b)$. By Taco-Taco with respect to adjacencies A, B and C, D, $\lambda_f(a,c) = -\lambda_f(b,d)$. By Pleat-Consistency on

A, C, E, $\lambda_f(a,c) = \lambda_f(c,e)$. By Pleat-Consistency on B, D, F, $\lambda_f(b,d) = \lambda_f(d,f)$. So $\lambda_f(c,e) = -\lambda_f(d,f)$. By Taco-Taco with respect to adjacencies C, D and E, F, $\lambda_f(c,f) = -\lambda_f(d,e)$. By Taco-Taco with respect to adjacencies C, E and D, F, $\lambda_f(c,d) = \lambda_f(e,f)$. Thus because $\lambda_f(a,b) = \lambda_f(e,f)$, the variable on the left has the same value as the one on the right. Alternatively if $\lambda_f(a,b) = -\lambda_f(c,d)$, the same series of arguments yields that $\lambda_f(c,d) = -\lambda_f(e,f)$, so $\lambda_f(a,b) = \lambda_f(e,f)$. Thus if global flat foldability holds, opposite variables are equal. Now assume that opposite variables are equal. The M/V assignment in Figure 2-4 completely induces λ_f, along with consistency and transitivity. The path shown is a linear order on the faces satisfying global layer ordering. Further, every other assignment of variables can be represented by a reflection of this crease pattern. □

Lemma 2.1.4. *The unassigned split gadget is a globally flat foldable crease pattern if and only if its three variables are equal.*

Proof. Refer to Figure 2-4. Assume global flat foldability. Let A and B be the faces containing points a and b respectively. The region highlighted in the figure and A must satisfy Path-Consistency, so $\lambda_f(a,b) = \lambda_f(a,c)$. Since the crease pattern is symmetric, $\lambda_f(b,a) = \lambda_f(b,c)$. Then, by antisymmetry, $\lambda_f(a,b) = \lambda_f(c,b)$, and therefore all variables are equal. Now assume all variables are equal. The path shown in Figure 2-4 is a linear order on the faces satisfying global layer ordering. Further, every other assignment of variables can be represented by a reflection of this crease pattern. □

Lemma 2.1.5. *The clause gadget is a globally flat foldable crease pattern if and only if its three variables are not all equal.*

Proof. Refer to Figure 2-4. Assume for contradiction the clause gadget is global flat foldable and all variables are equal. By consistency $\lambda_f(a,b) = \lambda_f(b,c) = \lambda_f(c,a)$. By transitivity, $\lambda_f(a,b) = \lambda_f(a,c)$. By antisymmetry, $\lambda_f(a,b) = -\lambda_f(c,a)$, a contradiction. Thus the variables are not all equal. Now assume all variables are not all equal. The paths shown in Figure 2-4 are linear orders on the faces satisfying global layer ordering. Further, every other assignment of variables can be represented by the negation of one of these (M/V) assignments. □

Theorem 2.1.2. UNASSIGNED-FLAT-FOLDABILITY *is NP-complete, even for box pleated crease patterns.*

Proof. Given λ_f as our certificate, we can check in polynomial time whether it satisfies all conditions for global flat foldability, therefore UNASSIGNED-FLAT-FOLDABILITY is in NP. By Lemma 2.1.3, Lemma 2.1.4, and Lemma 2.1.5, UNASSIGNED-FLAT-FOLDABILITY can simulate the SCN-SATISFIABILITY gadgets. It remains to check if the gadgets can be consistently connected. Let the width of a variable be the distance between its parallel creases. The crossover gadget connects variables of the same width while the clause and split gadgets both connect variables whose ratios differ by a factor of $\sqrt{2}$. Setting the width of one variable in any gadget induces the width of the other variables in the gadget. Fixing the width of one variable in the Complex Clause Gadget (Figure 2-3), a consistent unique width for all other variables is induced, resulting in the same width for each variable intersecting a left or right edge. Therefore, by Theorem 2.1.1, UNASSIGNED-FLAT-FOLDABILITY is NP-Hard. □

2.1.5 Assigned Crease Patterns

In this section we present gadgets simulating the elemental SCN gadgets with assigned crease patterns. They are shown in Figure 2-5.

We define a variable gadget as a set of parallel creases placed close together having a direction and a crease assignment as shown in Figure 2-5. By Taco-Tortilla, $\lambda_f(a,c) = \lambda_f(b,c) = \lambda_f(a,d) = \lambda_f(b,d)$, so, local to the gadget, it has exactly two globally flat foldable states. We say the variable is *True* if the faces to the right of the variable direction are above the faces to left ($\lambda_f(a,c) = 1$), and *False* otherwise.

Lemma 2.1.6. *The assigned crossover gadget is a globally flat foldable crease pattern if and only if opposite variables are equal.*

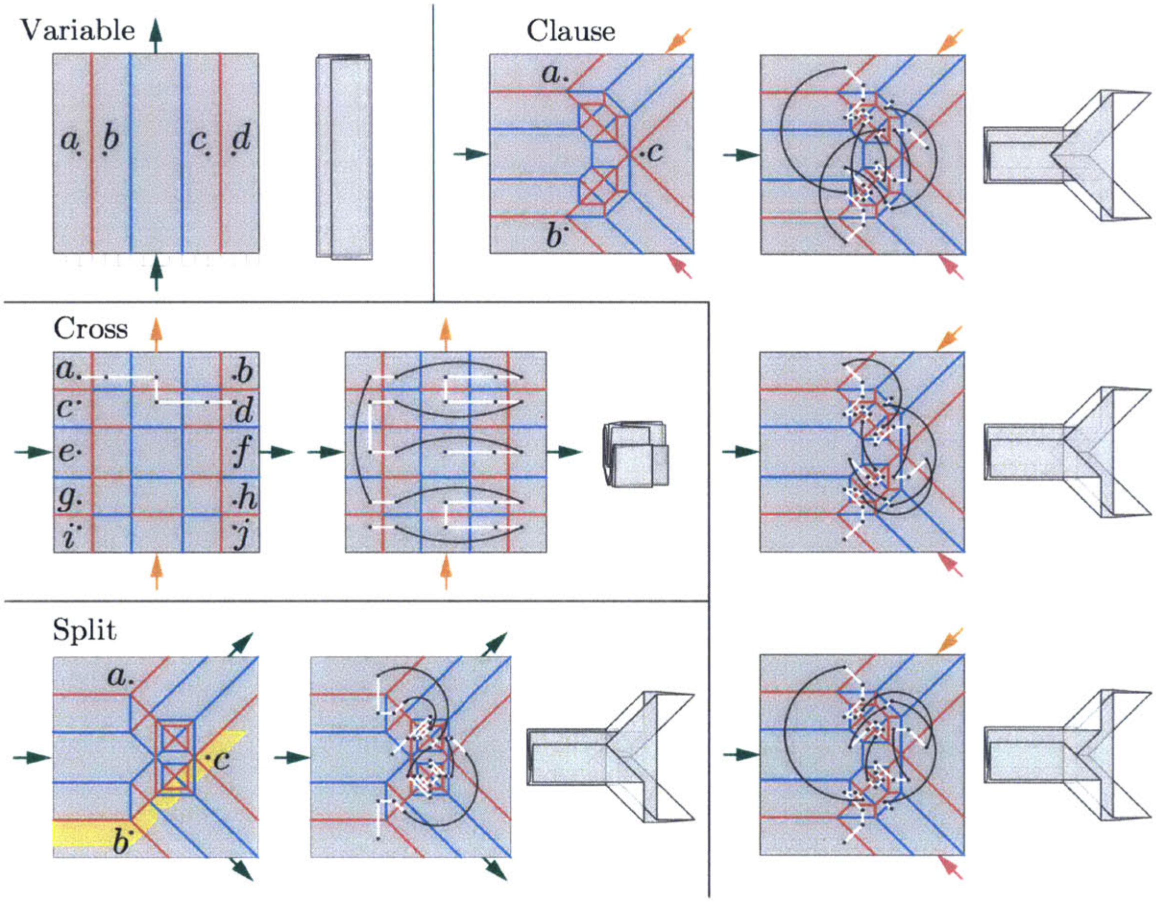


Figure 2-5: Elemental SCN Gadgets simulated with assigned crease patterns.

Proof. Refer to Figure 2-5. Assume global flat foldability. Let A, B, C, D be the maximal subsets of the faces respectively containing points a, b, c, d such that every pair strictly overlap. By transitivity on subset of λ_f induced by the M/V assignment shown, $\lambda_f(a, d) = \lambda_f(b, c) = -1$. By Taco-Taco with respect to adjacencies A, C and B, D, $\lambda_f(a, b) = -\lambda_f(c, d)$. Repeating this argument for adjacent rows of faces all the way down implies $\lambda_f(a, b) = -\lambda_f(c, d) = \lambda_f(e, f) = -\lambda_f(g, h) = \lambda_f(i, j)$. Thus, the variable on the top edge of the gadget has the same value as the one on the bottom. First assume $\lambda_f(g, a) = \lambda_f(a, b)$. Then previous implications imply $\lambda_f(g, a) = -\lambda_f(g, h)$. By transitivity and antisymmetry, $\lambda_f(g, a) = \lambda_f(h, b)$. Thus, the variable on the left side of the gadget has the same value as the one on the right. Alternatively, assume $-\lambda_f(g, a) = \lambda_f(a, b)$ so $\lambda_f(c, i) = \lambda_f(d, c)$. Then previous implications imply $\lambda_f(c, i) = \lambda_f(i, j)$. By transitivity and antisymmetry, $\lambda_f(c, i) = \lambda_f(d, j)$. Thus, the variable on the left side of the gadget has the same value as the one on the right. So, if globally flat foldable, opposite variables are equal. Now assume that opposite variables are equal. One can fix a unique λ_f by choosing a subset of λ_f in addition to the subset induced by the M/V assignment and consistency. The path shown in Figure 2-5 is a linear order on the faces satisfying global layer ordering. Further, every other assignment of variables can be represented by a reflection of this crease pattern. □

Lemma 2.1.7. *The assigned split gadget is a globally flat foldable crease pattern if and only if its three variables are equal.*

Proof. Refer to Figure 2-5. Assume global flat foldability. Let A and B be the faces containing points a and b respectively. The region highlighted in the figure and A must satisfy Path-Consistency, so $\lambda_f(a, b) = \lambda_f(a, c)$. Since the crease pattern is symmetric, $\lambda_f(b, a) = \lambda_f(b, c)$. Then, by antisymmetry, $\lambda_f(a, b) = \lambda_f(c, b)$, and therefore all variables are equal. Now assume all variables are equal. The path shown in Figure 2-5 is a linear order on the faces satisfying global layer ordering. Further, any other assignment of variables can be attained by a reflection. □

Lemma 2.1.8. *The assigned clause gadget is a globally flat foldable crease pattern if and only if its three variables are not all equal.*

Proof. Refer to Figure 2-5. Assume for contradiction the clause gadget is global flat foldable and all variables are equal. By consistency $\lambda_f(a,b) = \lambda_f(b,c) = \lambda_f(c,a)$. By transitivity, $\lambda_f(a,b) = \lambda_f(a,c)$. By antisymmetry, $\lambda_f(a,b) = -\lambda_f(c,a)$, a contradiction. Thus the variables are not all equal. Now assume all variables are not all equal. The paths shown in Figure 2-5 are linear orders on the faces satisfying global layer ordering. Further, any other assignment of variables can be attained by reversing the arrows in the figure. □

Theorem 2.1.3. ASSIGNED-FLAT-FOLDABILITY *is NP-complete, even for box pleated crease patterns.*

Proof. Given λ_f as our certificate, we can check in polynomial time whether it satisfies all conditions for global flat foldability and if it is consistent with the crease assignment, therefore ASSIGNED-FLAT-FOLDABILITY is in NP. By Lemma 2.1.6, Lemma 2.1.7, and Lemma 2.1.8, ASSIGNED-FLAT-FOLDABILITY can simulate the SCN-SATISFIABILITY gadgets. It remains to check if the gadgets can be consistently connected. Let the width of a variable be the distance between its two parallel mountain creases. By the same argument as in the proof of Theorem 2.1.2, widths of variables can be assigned consistently. Therefore, by Theorem 2.1.1, ASSIGNED-FLAT-FOLDABILITY is NP-Hard. □

2.1.6 Generating Instances

We wrote a program to automatically generate crease pattern constructed in the reductions presented. The program was written in coffeescript and can be found at http://jasonku.scripts.mit.edu/box-hard. Figure 2-6 and Figure 2-7 show crease patterns automatically generated by the software.

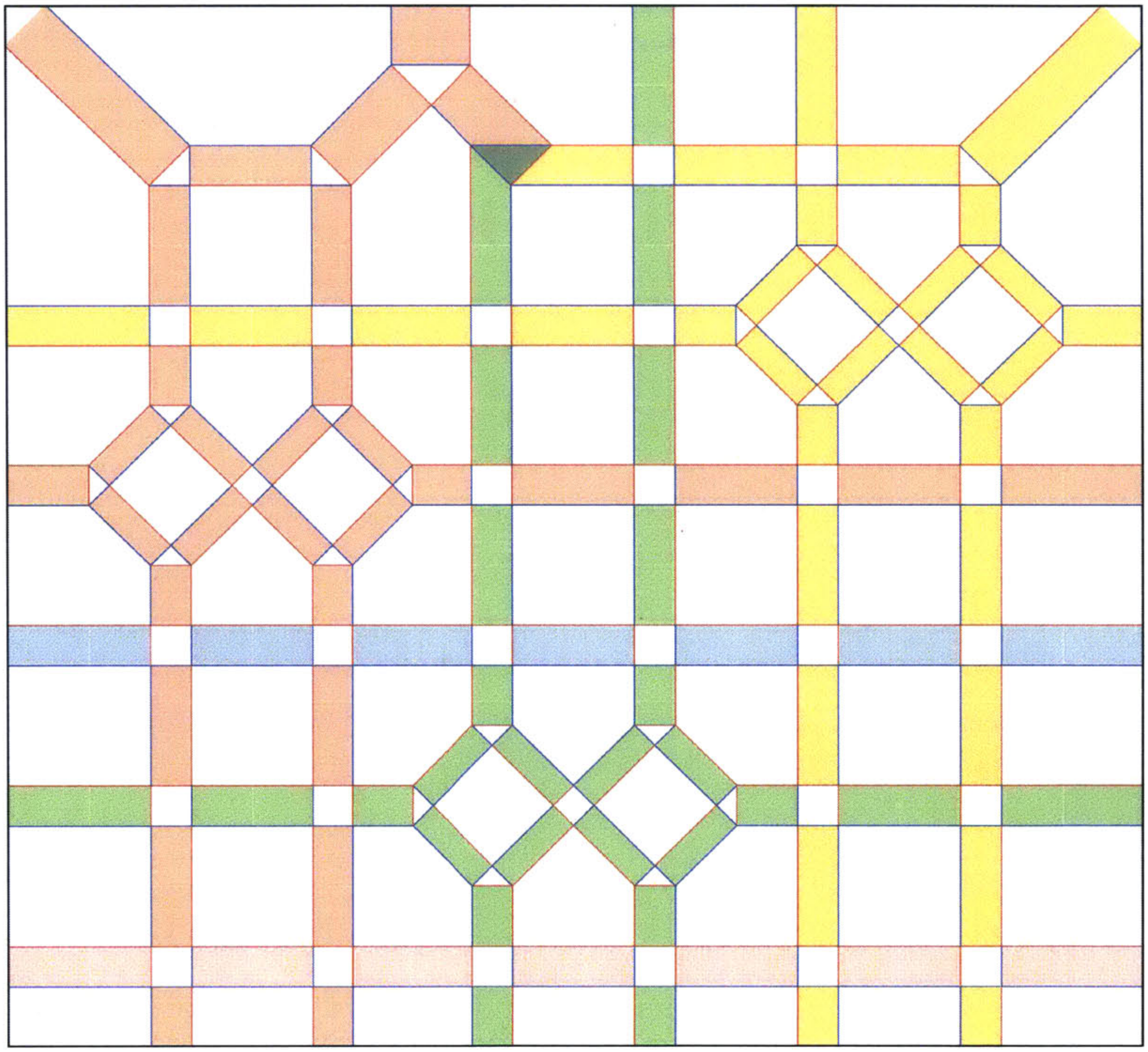


Figure 2-6: A crease pattern generated by our software for an unassigned Complex Clause Gadget. The gadget relates the yellow, red, and green variables with a satisfying M/V assignment, (yellow = True, red = False, green = false).

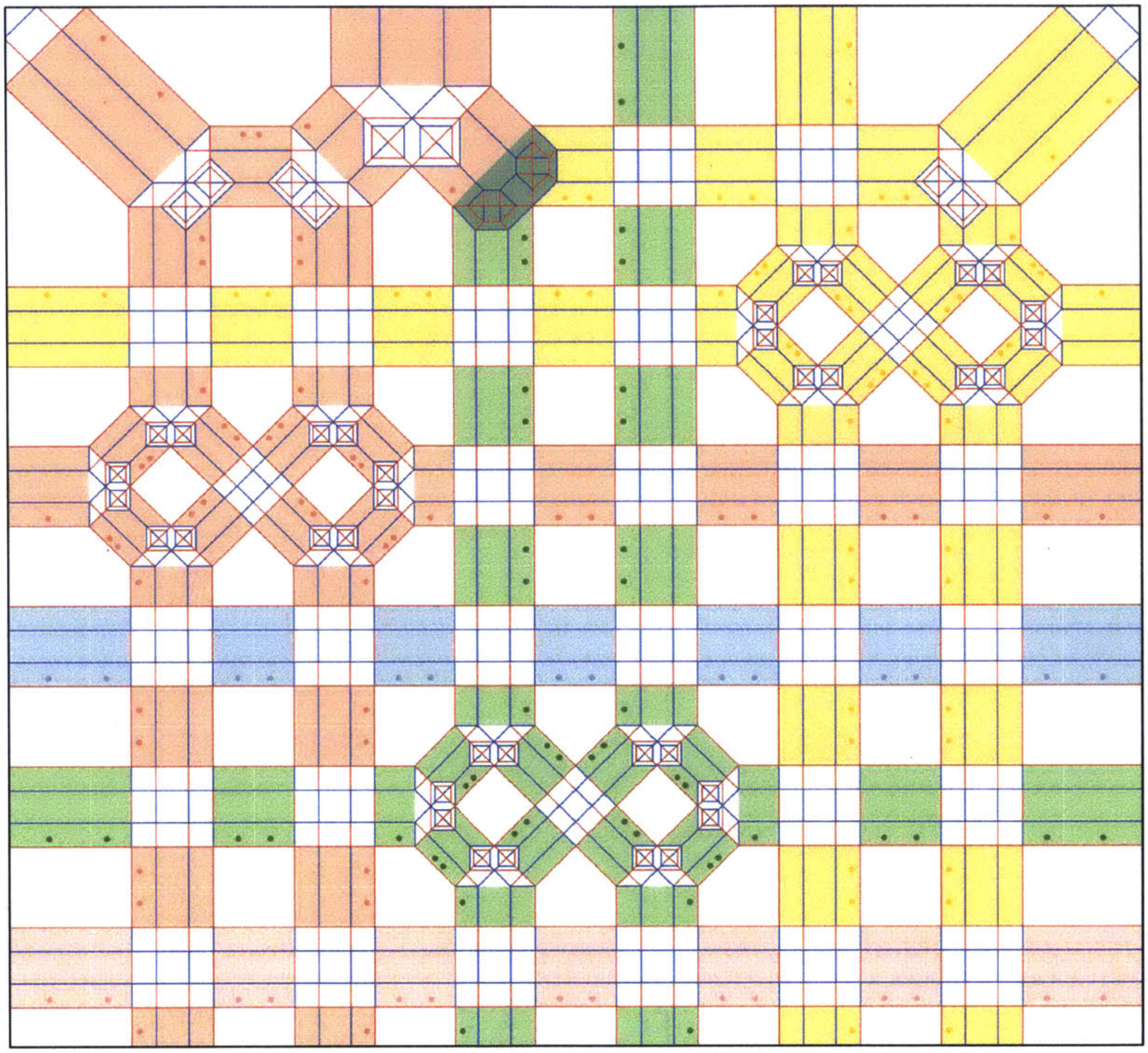


Figure 2-7: A crease pattern generated by our software for an assigned Complex Clause Gadget. The gadget relates the yellow, red, and green variables with a satisfying M/V assignment, (yellow = True, red = False, green = false). Dots indicate the layer that folds below.

2.1.7 Remarks

Table 2.1 overviews our results and open problems. We proved Unassigned-Flat-Foldability and Assigned-Flat-Foldability are NP-complete, even for box pleated crease patterns containing no more than 9 and 25 layers respectively. Are these problems still hard for even more restricted inputs? The computational complexity of Assigned-Flat-Foldability is still open when the crease pattern is a $m \times n$ square grid called a map [4]. Between box pleating and map folding is orthogonal folding, with crease patterns restricted to orthogonally aligned creases.

	General	**Box Pleating**	Orthogonal	Map
Unassigned	Hard [9]	**Hard (Ours)**	Poly [4]	Always True
Assigned	Hard [9]	**Hard (Ours)**	Open	Open

Table 2.1: Overview of **our results** and open problems. 'Hard' and 'Poly' designate problems that are NP-complete or solvable in polynomial time respectively.

2.2 Simple Folding is Hard

As discussed in the previous section, perhaps the most researched subset of origami studies flat foldings—folded states that lie in the plane, with multiple non-overlapping layers.

In this section, we study *simple foldability*, deciding whether a 2D crease pattern can be folded by a sequence of simple folds. Informally, a simple fold can only rotate paper around a single rotation axis before returning the paper back to the plane. This restriction is motivated by practical sheet-metal bending, where a single robotic tool can fold the sheet material at once. We build on the work of Arkin et al. [4]. They introduce many models of simple folds, proving that deciding simple foldability is weakly NP-complete for some of them, and that simple foldability can be solved in polynomial time for rectangular paper with paper-aligned orthogonal creases. We abbreviate this restriction on the input (rectangular paper and paper-aligned orthogonal creases) as (⊞) crease patterns, and will abbreviate other restrictions similarly. We also introduce a new model of simple folding, namely the infinite simple folds model where simple folds must fold at least one layer everywhere the paper intersects the fold axis, which is a more accurate model of existing machine tools like mechanical brakes.

We prove strong NP-completeness for every model proved weakly NP-complete in [4], namely that simple foldability is hard for:

1. orthogonal paper with paper-aligned orthogonal creases (abbreviated ⊞) with crease assignment in the one-layer, some-layers, and all-layers models, even to approximate the number of possible simple-folds; and
2. square paper with paper-aligned creases at multiples of 45° (abbreviated ⊠) with crease assignment in the some-layers and all-layers models.

Additionally we prove strong NP-completeness deciding simple foldability of:

3. ⊠ crease patterns without crease assignment in the some-layers and all-layers models; and
4. ⊞ crease patterns with or without crease assignment in the infinite one-layer and some-layers models.

We also point out some errors in the unassigned weakly NP-complete reduction in Arkin et al. folding orthogonal polygons with assigned unconstrained creases, but we do not comment further as the result is subsumed by result (3) above. In the last section, we extend the polynomial-time result from [4] to the infinite simple folds models, proving the infinite and non-infinite models are equivalent for ⊞ crease patterns. Table 2.2 shows the computational complexity of simple-foldable decidability in various models.

2.2.1 Definitions

In general, we are guided by the terminology laid out in [4], though for this paper we restrict our discussion to folding two-dimensional paper. We will operate in $\mathbb{R}^3$ containing a folding plane $\mathbb{P}$ congruent to $\mathbb{R}^2$ with a surface normal vector $\hat{n}$. A

Model	**Assigned**			**Unassigned**		
	⧉	⊠	⊞	⧉	⊠	⊞
One-layer	weak → **strong**	open	poly	open	open	poly
Some-layers	weak → **strong**	weak → **strong**	poly	open	**strong**	poly
All-layers	weak → **strong**	weak → **strong**	poly	open	**strong**	poly
Inf. One-layer	**strong**	open	**poly**	**strong**	open	**poly**
Inf. Some-layers	**strong**	open	**poly**	**strong**	open	**poly**
Inf. All-layers	open	open	**poly**	open	open	**poly**

Table 2.2: Computational complexity of simple folding problems, either open, solvable in polynomial time (poly), or strongly/weakly NP-complete (strong/weak). **Bold** results are new in this paper. Rows list simple folding models while the columns list restrictions on the input: orthogonal paper/orthogonal creases (⧉), square paper/45°creases (⊠), or rectangular paper/orthogonal creases (⊞) .

two-dimensional *paper* is a connected polygon in $\mathbb{P}$, possibly with holes. We denote the boundary of P by ∂P. We call the side of a paper pointing in the $\hat{n}$ direction the *top* and the opposite side the *bottom*. A *crease* is a line segment on a paper. A *crease pattern* (P, Σ) is a paper P and a set of creases Σ contained in the paper, no two of which intersect except at a common endpoint. A *facet* of a crease pattern is a connected open set in $P \setminus \Sigma$ whose boundary is a subset of $\partial P \cup \Sigma$. Two crease pattern facets are *adjacent* if their boundaries share a common crease.

A *flat fold isometry* (P, Σ, f) is a crease pattern (P, Σ) together with an isometric embedding f of the paper into $\mathbb{P}$ such that (1) each facet of the crease pattern is mapped to a congruent copy, (2) connectivity is preserved between facets and creases, and (3) for every pair of adjacent facets, exactly one of the facets is reflected in the embedding. If a crease pattern has a flat fold isometry, we call the crease pattern *locally flat-foldable*, which is checkable in polynomial time [9]. If a flat folding exists it is unique, with Σ induced by f, so we will use (P, Σ, f), (P, f), or (P, Σ) interchangeably to denote flat fold isometries and locally flat-foldable crease patterns. We denote the preimage of $U \subset f(P)$ as $f^{-1}(U) \subset P$.

A *flat folding* (P, Σ, f, λ) is a flat fold isometry (P, Σ, f) together with a *layer ordering* [39], a partial ordering λ on crease pattern facets such that (1) points in the same facet have the same layer order, (2) facets whose images intersect in the flat fold isometry have different layer order, and (3) for any crease, no facet whose image intersects the crease's image in the flat fold isometry is between the facets adjacent to the crease in the layer ordering. We say a point with higher layer order than another is *above*, with the other point being *below*. If a crease pattern has a flat folding, we call the crease pattern *globally flat-foldable.*

The *flat-foldability decision problem* takes as input a locally flat-foldable crease pattern and asks if it is globally flat foldable. If no other information is given, the problem is called *unassigned.* A common variant of the decision problem also provides in the input an assignment $\alpha : \Sigma \to \{\mathrm{M},\mathrm{V}\}$ of the creases to either mountain or valley, and the question asks if a flat folding exists satisfying the assignment according to the following definitions. Given a flat folding, we call a crease a *mountain* (M) if, of the two facets adjacent to the crease, the facet with lower layer order is reflected in the flat fold isometry; and a *valley* (V) if the facet with higher layer order is reflected. This definition adheres to the intuition that a valley folds with the top surface inside while a mountain folds with the top surface outside. Arbitrarily assigning mountain or valley to the creases of a flat fold isometry may be consistent with zero, one, or multiple flat foldings. If α is given, the decision problem is called *assigned.*

2.2.2 Description of a Simple Folding

A *simple folding* $(P, \Sigma_2, f_2, \lambda_2)$ of an input flat folding, $(P, \Sigma_1, f_1, \lambda_1)$, is itself a flat folding parameterized by a *fold axis* (a line $\ell(a, b \in \mathbb{P}) = \{ax + b | x \in \mathbb{R}\}$) and a *folded region* (a subset $U \subsetneq P$) satisfying the following conditions.

(1) Points on the boundary of the folded region are either in the boundary of the paper or the preimage of the fold axis, (2) the folded region does not contain points in the preimage of the fold axis but does contain the rest of its boundary points, (3) points not in the folded region are unchanged, (4) everything in the folded region moves to a reflected point across the fold axis, (5) the folded region is appropriately above or below points not in the folded region in the input layer ordering, (6) the output layer ordering of the folded region is exactly the opposite the input layer

ordering, (7) points in the folded region lie appropriately above or below points not in the folded region in the output layer ordering, and (8) the creases of the new flat folding contain the creases of the old one. Formally:

1. $\partial U \subset \partial P \cup f_1^{-1}(\ell \cap f_1(P))$;
2. $U \cap f_1^{-1}(\ell \cap f_1(P)) = \emptyset$ and $\partial U \setminus f_1^{-1}(\ell \cap f_1(P)) \subset U$;
3. $f_2(p) = f_1(p)$ for $p \in P \setminus U$ and $\lambda_2(p) = \lambda_1(p)$ for $p \in P \setminus (U \cup \Sigma_2)$;
4. $f_2(u) = 2(f_1(u) + (f_1(u) - b) \cdot a) + f_1(u)$ for $u \in U$;
5. $\lambda_1(u)(>,<)\lambda_1(p)$ for $u \in U, p \in f_1^{-1}(f_1(u)) \setminus U$, $(u - b) \times a\,(>,<)\,0$;
6. $\lambda_2(u)(>,=,<)\lambda_2(v)$ if and only if $\lambda_1(u)(<,=,>)\lambda_1(v)$ for $u, v \in U$;
7. $\lambda_2(u)(>,<)\lambda_2(p)$ for $u \in U, p \in P \setminus U, \operatorname{sgn}((u - b) \times a)(>,<)\operatorname{sgn}((v - b) \times a)$;
8. and $\Sigma_1 \subsetneq \Sigma_2$.

A *simple fold* is then a rotation of a folded region in a flat folding about a fold axis back into the plane to form a new flat folding that is a simple folding as described above. Conditions (1) and (2) of a simple folding ensure the rotation is isometric during folding; conditions (3) and (4) ensure that folding occurs exactly in the folded region; condition (5) ensures that the paper does not intersect itself; conditions (6) and (7) ensure the layers of the output are consistent; and condition (8) ensures that existing creases do not unfold.

We define different models of simple folding that limit the choice of U. Let $L = \ell \cap f(P)$ be the intersection of fold axis ℓ and input flat folding $(P, \Sigma_1, f_1, \lambda_1)$, and let $\#_+(q) = |f^{-1}(q) \setminus (\partial P \cup \Sigma_1)|$, the number of foldable layers at $q \in L$. Then the function $\# : L \to \{0, \ldots, \#_+\}$ denotes the number of layers that are folded in a simple fold at every point along the fold axis, specifically $\#(q) = |(f^{-1}(q) \cap \partial U) \setminus (\partial P \cup \Sigma_1)|$ for $q \in L$. Table 2.3 defines our models of simple folding based on restrictions on # that limit the choice of folded region. Of particular interest is the infinite all-layers

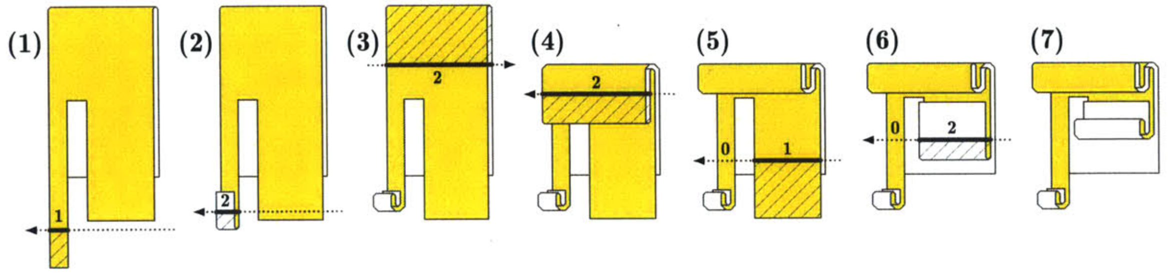


Figure 2-8: Example folding steps demonstrating the differences between simple folding models. L is a directed dotted line in the direction of a, U is textured, and the fold line $f^{-1}(L) \cap \partial U$ is a thick line with the number of layers # specified.

Simple Folding Model	**Restriction on # for $q \in L$**	Foldable Example Steps
Some-layers	no restriction	(1), (2), (3), (4), (5), (6)
One-layer	$\#(q) \in \{0, 1\}$	(1), (5)
All-layers	$\#(q) \in \{0, \#_+(q)\}$	(1), (2), (3)
Infinite Some-layer	$\#(q) \geq 1$	(1), (3), (4)
Infinite One-layer	$\#(q) = 1$	(1)
Infinite All-layers	$\#(q) = \#_+(q)$	(1), (3)

Table 2.3: Definitions for different models of simple folding according to restrictions on the number of layers that must be folded along the fold axis. Example steps are shown in Figure 2-8.

model which corresponds to folding everything on one side of the fold axis to the other side, a model which has practical applications in manufacturing.

If we put no restriction on the number of layers that may be folded in a simple fold, then we are in the *some-layers* simple folds model. If we require the folding of zero or one layer(s) in a simple fold, with $\#(q) \in \{0, 1\}$ for all $q \in L$, then we are in the *one-layer* simple folds model. If we require the folding of either zero or all layers, with $\#(q) \in \{0, |f^{-1}(q)|\}$ for all $q \in L$, then we are in the *all-layers* simple folds model. If we require the folding of exactly one layer, with $\#(q) = 1$ for all $q \in L$, then we are in the *infinite one-layer* simple folds model. If we require the folding of one or more layers, with $\#(q) \geq 1$ for all $q \in L$, then we are in the *infinite some-layers* simple folds model. Lastly, if we require the folding of all layers along the fold axis, with $\#(q) = |f^{-1}(q)|$ for all $q \in L$, then we are in the *infinite all-layers* simple folds model.

Given locally flat-foldable crease pattern (P, Σ, f), we say that it is *simply-foldable* or equivalently *flat-foldable via a sequence of simple folds* in some model, if the crease pattern can be folded by a sequence of m simple folds into a series of flat foldings $S_i = (P, \Sigma_i, f_i, \lambda_i)$ for $i \in \{1, \dots, m\}$ such that S_1 is the original unfolded paper with $\Sigma_1 = \emptyset$, each flat folding S_{i+1} is a simple folding of S_i, and $S_m = (P, \Sigma_m = \Sigma, f_m = f, \lambda_m)$ is a flat folding of the input.

If it is hard to decide simple-foldability, a natural question arises: how close can we estimate the number of possible simple folds that can be performed? Define MAXFOLD, the natural optimization version of the decision problem asking for the maximum number of simple folds that can be folded given a locally flat-foldable crease pattern (P, Σ), or formally, the maximum length sequence of simple folds to fold any simply-foldable crease pattern (P, Σ') with $\Sigma' \subset \Sigma$.

2.2.3 Orthogonal Paper/Orthogonal Creases

In this section we prove that the simple-foldability decision problem of an orthogonal piece of paper with a M/V assigned paper-aligned orthogonal crease pattern (⊞) is strongly NP-complete in the one-layer, some-layer, and all-layer models of simple folding. This result is the same as Theorem 6.3 from [4], but proves strong NP-completeness because we reduce from a strongly NP-complete problem. Additionally, we prove that it is hard even to approximate the associated natural optimization problem.

Theorem 2.2.1. *The assigned simple-foldability decision problem for orthogonal paper with paper-aligned orthogonal creases* ⊞ *is strongly NP-complete in the one-layer, some-layers, and all-layers models.*

Proof. The proof is by reduction from 3-PARTITION. Given an instance of 3-PARTITION with integers $A = \{a_1, \ldots, a_n\}$ to be partitioned into $n/3$ triples each with sum $\sum A/(n/3) = t$, construct an orthogonal polygon with M/V assigned paper-aligned orthogonal creases as shown in Figure 2-9 (the width of the polygon is one everywhere). We assume each a_i is close to $t/3$ and divisible by $2n$: if not, add a large number to each and multiply by $2n$ so that they are.

There are five main functional sections of the polygon, as shown in Figure 2.2.3. On the left is the Bar, a section whose convex hull is a $5 : 2\infty$ rectangle of paper without creases that is very long ($\infty = 10nt$). Attached to the middle of the Bar is a $\frac{5n}{3} + \frac{1}{2}$ long strip extending to the right which we call the Arm. The Staircase encodes the a_is in order as a series of steps with height equal to their value plus one. Step i contains two creases c_{2i-1}, c_{2i} that when both folded raise the Bar by exactly $2a_i$. The Wrapper section is a horizontal rectangle of length $2n/3$ with vertical valley creases d_i (d_1 being the right most crease) dividing the Wrapper into unit squares. The Cage on the right bounds a polygonal area whose the left vertical edge we call the Column.

The construction forces the Bar to wrap inside the Cage $n/3$ times, each time shifted up by distance $2t$ (note that ∞ is chosen large to ensure that the Staircase never intersects the Cage polygon while wrapping). To prove the claim, we first prove the Wrapper must fold its vertical creases in order from right to left. If this were not the case, then there exists some first crease d_i to be folded whose right neighbor d_{i-1} has not yet been folded. But d_i has at least two squares of unfolded paper to its

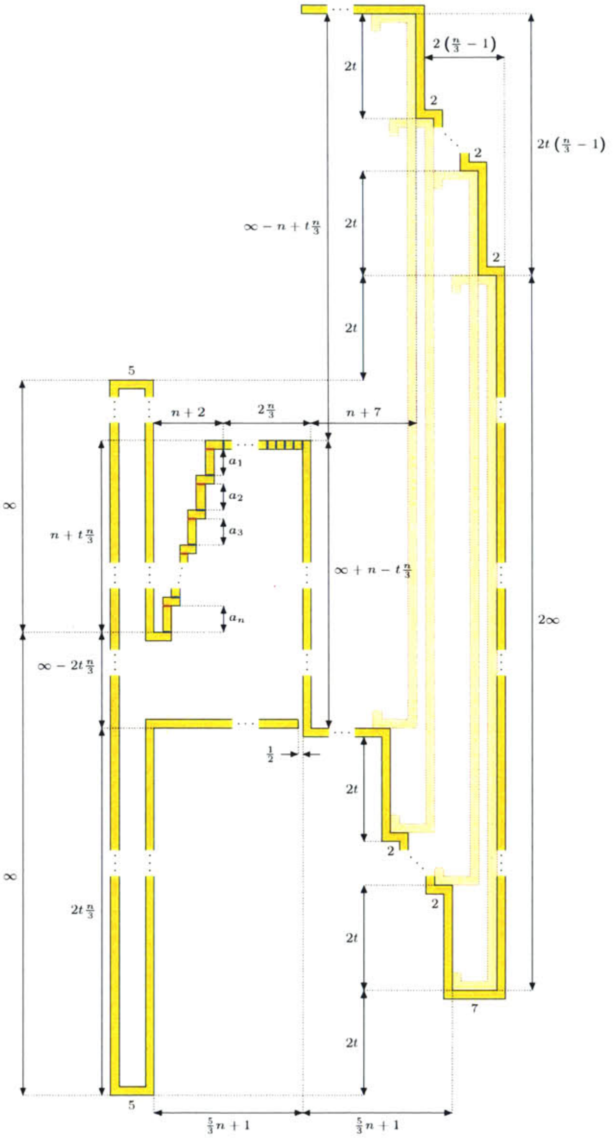


Figure 2-9: An orthogonal simple polygon with orthogonally aligned mountain-valley creases (drawn in red and blue respectively) constructed from an instance of 3-PARTITION that can be folded using simple folds if and only if the instance of 3-PARTITION has a solution.

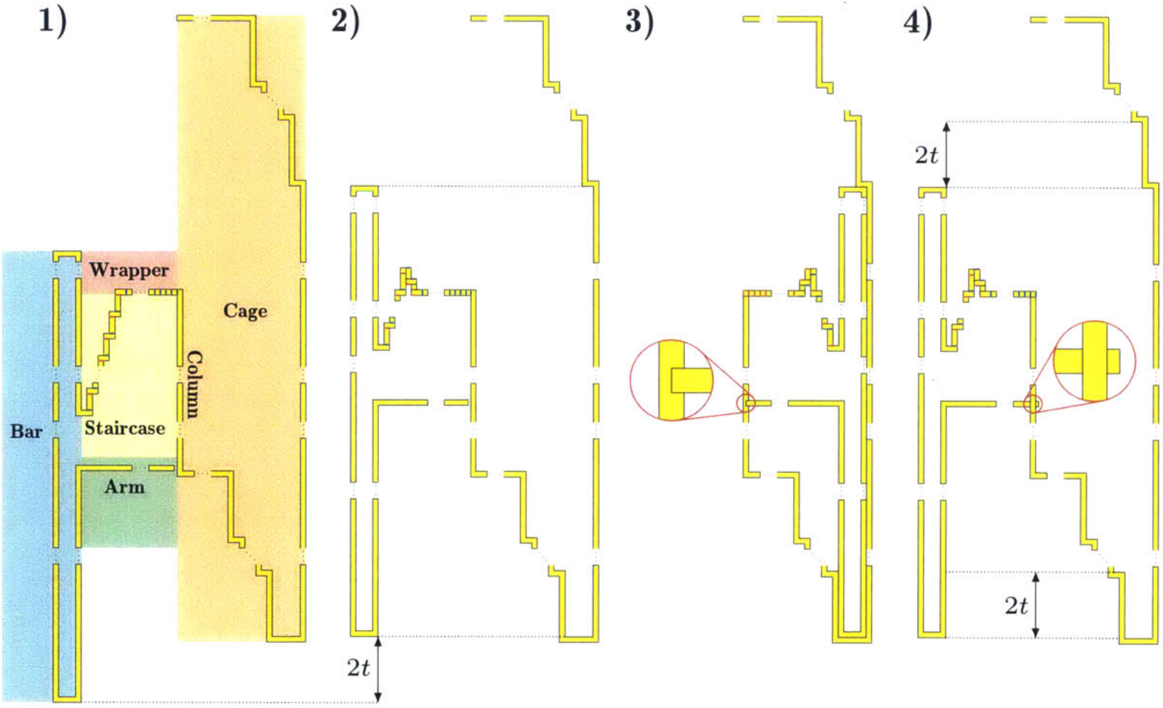


Figure 2-10: Process to check the Partition solution: 1) pleat variables to change height of bar by $2t$, 2) fold along the rightmost wrapper crease around the column, 3) fit the bar through the cage folding the bar to the left along the next wrapper crease, 4) repeat until $n/3$ triples adding to $2t$ have been checked.

left that will cover d_{i-1} when folded, making d_{i-1} impossible to fold using simple folds without violating the M/V assignment, contradicting our model. Because the Wrapper executes its folds from right to left, the Bar must pass through the Cage $n/3$ times sequentially from the rightmost slot to the leftmost, with each subsequent slot shifted up by $2t$.

If the 3-Partition instance has a positive solution, then the polygon has a simple folding: just pleat the creases associated with the a_is in one of the satisfying triples, then fold the Bar through the Cage along the next Wrapper creases, and repeat. Because all folds in the Wrapper are all valley, the Arm will go around the Column and never cross it. Further, if the polygon has a simple folding, the 3-Partition instance has a positive solution because the Staircase must be folded on both creases from exactly three a_is between each wrap. To prove this, all a_is are close to $t/3$ so in order to shift by $2t$, exactly three a_i sections must be flipped from their original orientation. Further, because each a_i is divisible by $2n$, no one unit section between a_is can flip if the total height is to raise by t, since t is also divisible by $2n$. So the a_is flipped at each step correspond to triplets of the 3-Partition instance that sum to t.

Folded in this way, each simple fold can be performed in the one-layer and some-layers models because the construction only ever folds through one layer of paper at a time. And because creases only ever exist in a single layer, the all-layers model also applies. The reduction is polynomial because the entire constructed polygon is bounded by a $30nt \times 4n$ rectangle. Lastly, the problem is in NP because given a certificate of the crease folding order, each fold can be simulated and checked in polynomial time. □

2.2.4 Inapproximability

The optimization version of the decision problem is even hard to approximate.

Theorem 2.2.2. *Given an orthogonal paper with paper-aligned orthogonal creases* ⊞ *admitting a maximum sequence of m simple folds, approximating* MaxFold *to* *within a factor of $m^{1-\varepsilon}$ for any constant $\varepsilon > 0$ is NP-complete in the some-layers and all-layers models.*

Proof. Construct a crease pattern similar to Figure 2-9, but with the Wrapper modified in the following way: add δ horizontal lines of creases all the way through the Wrapper, with each horizontal line composed of $2n/3+1$ collinear creases alternating M/V assignment in each section between vertical creases, splitting each of the $2n/3$ vertical creases into $\delta + 1$ collinear vertical valley creases.

For a positive instance of 3-PARTITION, the proof of Theorem 2.2.1 implies that the $8n/3$ original creases may be folded as simple folds, then allowing δ more simple folds to be performed by folding along each line of horizontal of creases from top to bottom in the some-layers and all layers models. None of the added horizontal creases can be folded before all vertical creases in the Wrapper are folded due to M/V alternation along the line. This construction is thus simple-foldable via a sequence of $m = 8n/3 + \delta$ simple folds.

For a negative instance of the 3-PARTITION problem, there exists at least one line of vertical Wrapper creases that cannot be folded, reducing the number of possible simple folds to strictly less than $8n/3$.

Setting $\delta = (8n/3)^{1/\varepsilon} - 8n/3$, Theorem 2.2.1 implies it is NP-hard to distinguish the case where m folds are possible from the case where at most $8n/3 = m^{\varepsilon}$ are possible. The reduction is polynomial since both δ and m are $O(n^{1/\varepsilon})$ for constant ε. □

2.2.5 Assigned Square Paper/45° Creases

Arkin et al. adapt their PARTITION reduction to square paper with M/V assigned paper-aligned creases at multiples of 45° (⊠) by constructing an approximation of their orthogonal construction from a square. Unfortunately their modification cannot be applied to our 3-PARTITION reduction in the all-layers model because their construction requires folds along the long construction end which may (will) overlap other parts of the paper during construction.

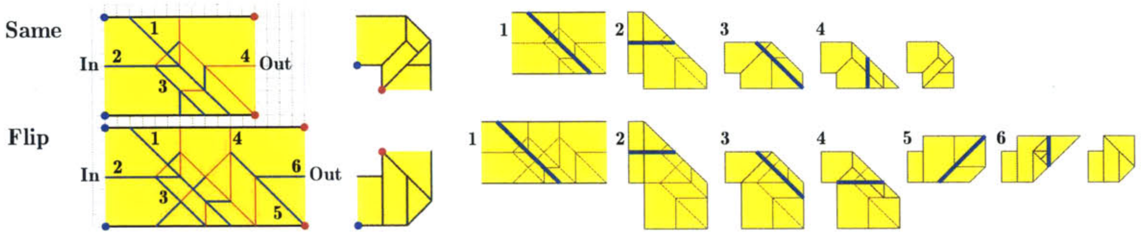


Figure 2-11: Turn gadgets for assigned case. Red/blue lines represent the M/V assignment.

Instead, we use a similar idea to construct an orthogonal polygon approximation from a square but with a different turn gadget that enforces the order of construction while only making folds local to the gadget that works in both the some-layers and all-layers models.

Theorem 2.2.3. *The assigned simple-foldability decision problem for square paper with paper-aligned creases at multiples of* 45° ⊠ *is strongly NP-complete in the some-layers and all-layers models.*

Proof. The proof is by reduction from the decision problem in Theorem 2.2.1. Given such an orthogonal polygon with M/V assigned paper-aligned orthogonal creases, we construct a crease pattern on a square that folds using simple folds if and only if the original orthogonal crease pattern is simply-foldable.

We start by constructing a long rectangle from the starting square of appropriate aspect ratio in the same way as [4], double the width of the orthogonal polygon we want to create. Then we use turn gadgets to shape the long rectangle into the target orthogonal polygon. Figure 2-11 depicts crease patterns for our turn gadgets, Same and Flip, along with drawings depicting their valid flat foldings. We call creases located on the horizontal center line halfway between the edges of the paper *axial* creases. These crease patterns have the property that the axial crease extending the right edge (the output) cannot be folded unless all creases in the gadget have already been folded.

When folded, both gadgets align the edges of the original long rectangle to one side. Having both the Same and Flip gadgets allows us to combine them in one long strip to turn right or left no matter which side the original edges are on. The Same gadget turns the paper to the same side as the original edges, while the Flip gadget turns the paper to the other side. If chained in a sequence, turning in the same direction as the previous turn necessitates a Same gadget, while the Flip gadget turns the paper in the opposite direction.

The construction is as follows. We trace the path of the target orthogonal polygon starting at the cage end. Wherever a turn is needed, apply the appropriate turn gadget. The creases of the target crease pattern are overlaid to be foldable only after the appropriate section has been folded in half. If the orthogonal polygon is simply-foldable, we can then fold the remaining creases.

Now we prove the orthogonal polygon can be folded if the square crease pattern is simply-foldable. Before any section can be folded along axial creases, all creases behind the axial crease must have already been folded. The gadgets can be folded using only valley folds, so the paper will never self intersect. Further, creases local to a turn gadget do not overlap other paper because gadgets are far from each other. In particular, no crease of the target crease pattern may fold before the cage is constructed. Once the cage has been constructed, no Wrapper crease may fold until the Bar has been constructed completely because any uncreased paper will be too large to fit through the cage.

The reduction is polynomial because the side of the input square is bounded by $O(nt)$ and the number of creases is bounded by $O(n^2)$. Lastly, the problem is in NP because given a certificate of the crease folding order, each fold can be simulated and checked in polynomial time. □

2.2.6 M/V Unassigned Square Paper/45° Creases

M/V unassigned crease patterns are naturally less restrictive than M/V assigned crease patterns. This freedom can make collision avoidance easier, providing a choice folding direction at each crease. However when proving hardness for M/V unassigned

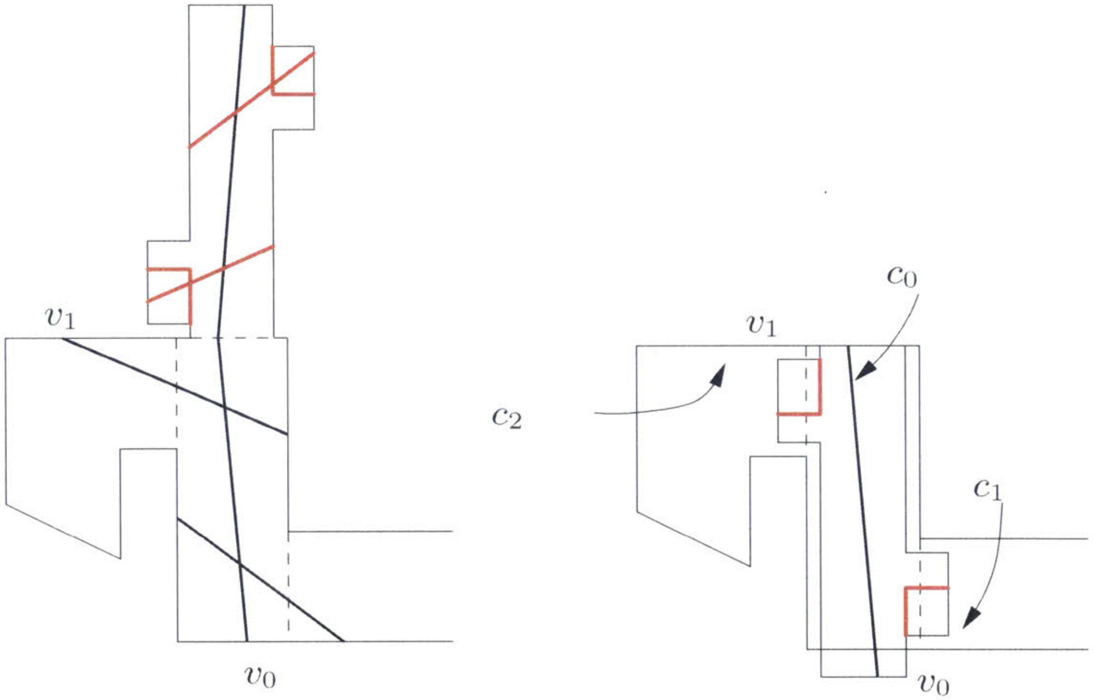


Figure 2-12: Figure 18 from [4]. Corrections marked in red creating reflections of c_1 and c_2 on the covering flap, and trimming the covering flaps so that c_1 and c_2 do not intersect v_0 or v_1 within the covering flap.

crease patterns, one cannot use crease direction to enforce fold ordering or layering and must restrict them using other techniques. Arkin et al. provide a weakly NP-hard reduction for orthogonal polygons with unconstrained creases without crease assignment in Theorem 7.1, but their proof has two errors.

The first error in the proof of Theorem 7.1 in [4] is that Arkin et al. claim that their reduction for the M/V assigned case can be used directly to prove hardness of the M/V unassigned all-layers model, saying, "in the all-layers case, as soon as two layers of paper overlap they are 'stuck' together." However, this claim is not true under their definition of the all-layers model.

The second error is a fixable problem in the creases shown in Figure 18. Their construction modifies their PARTITION reduction by adding pleats to force the folding direction of creases v_1 and v_2, claiming the added cross pleats must fold first to enforce

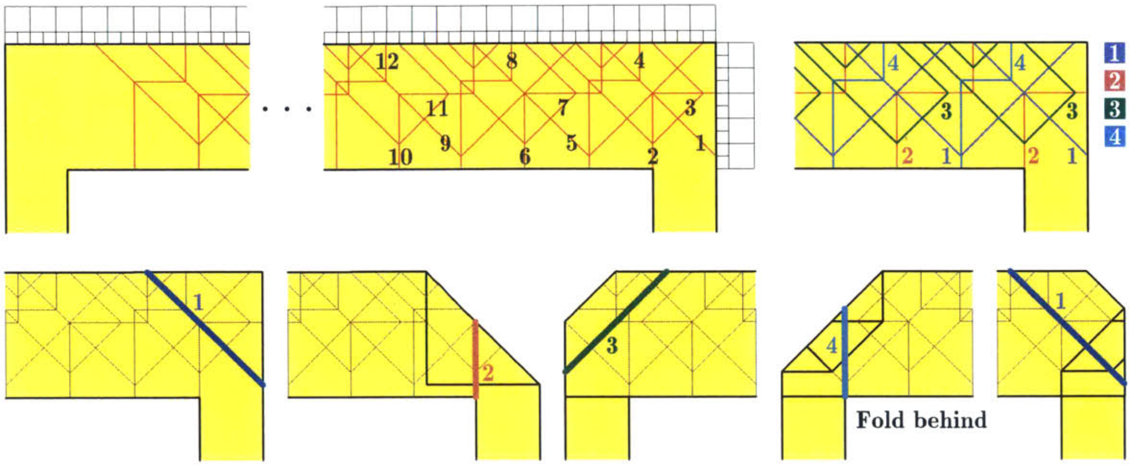


Figure 2-13: (Top-left) Crease pattern for the Wrapper in the unassigned model. Red lines show unassigned creases. (Top-right) Creases are colored according to their folding order. (Bottom) Folding sequence showing the creases that are being folded.

v_0 and v_1 to fold in the same direction. However, pleating c_0 before c_1 and c_2 locks the latter two creases to paper containing no creases, preventing them from ever folding. Adding mirrored creases on the cover fixes this problem. Further, the position of creases c_1 and c_2 lock the layers containing v_0 and v_1 to overlapping uncreased paper meant to enforce folding direction. Trimming problematic extra paper can fix the proof. We do not elaborate further as we prove stronger results that subsume Theorem 7.1, namely Theorems 2.2.4 and 2.2.5.

Theorem 2.2.4. *The unassigned simple-foldability decision problem for orthogonal paper with paper-aligned creases at multiples of* 45° *is strongly NP-complete in the some-layers and all-layers models.*

Proof. The proof of Theorem 2.2.1 still holds using the same construction with unassigned crease patterns except for two points: (1) the argument ensuring that the creases of the Wrapper fold in order from right to left does not apply without crease assignment; and (2) the argument ensuring that the bar folds through the cage each time requires every vertical fold in the Wrapper to either be all mountain or all valley.

To fix problem (1), we modify the Wrapper paper to be two units tall and replace the Wrapper creases with the creases shown in Figure 2-13. These creases have the property that the new creases $\Sigma_{i+1} \setminus \Sigma_i$ added in any sequence of simple folds resulting in a simple folding of the Wrapper is uniquely defined in the all-layers model, namely each simple fold S_i can only be folded if all simples folds S_j for $j < i$ have already been folded. In the some-layers model, the order on simple folds is not quite unique since some strict subset of creases in some of the $\Sigma_{i+1} \setminus \Sigma_i$ above may fold out of order, but it remains that for any crease in Σ_i to fold, some nonempty subset of Σ_j must have already been folded for $j < i$, enforcing the new Wrapping creases to fold in order from right to left. The ordering is given in the figure and follows from the observation that a simple fold can only occur when the subset of creases to be folded divides the paper, with creases collinear in the flat folding.

To fix (2), we must ensure that the vertical Wrapper creases are either all mountain or all valley in any flat folding reachable as a sequence of simple folds. The Arm in the original construction (not useful for the original theorem) is included to enforce this requirement. After the rightmost vertical Wrapper crease has been fold first, the Arm will overlap the Column. Since the order is enforced by (1), the second vertical Wrapper crease will fold next while to the right of the Column. If it folds with assignment opposite the first, the Arm would intersect the Column contradicting the simplicity of the fold. This argument holds inductively for the remainder of the vertical Wrapper creases, so they must all fold with the same assignment in any sequence of simple folds.

Having addressed these two problems, the arguments of Theorem 2.2.4 directly apply under both the some-layers and all-layers models, since the order and assignment of the Wrapper creases are forced in both models. □

Theorem 2.2.4 goes beyond Theorem 7.1 from [4] by both proving a stronger notion of hardness and restricting creases to only multiples of 45°. The following is an even stronger result, showing that the problem is still hard even when the orthogonal polygon is a square.

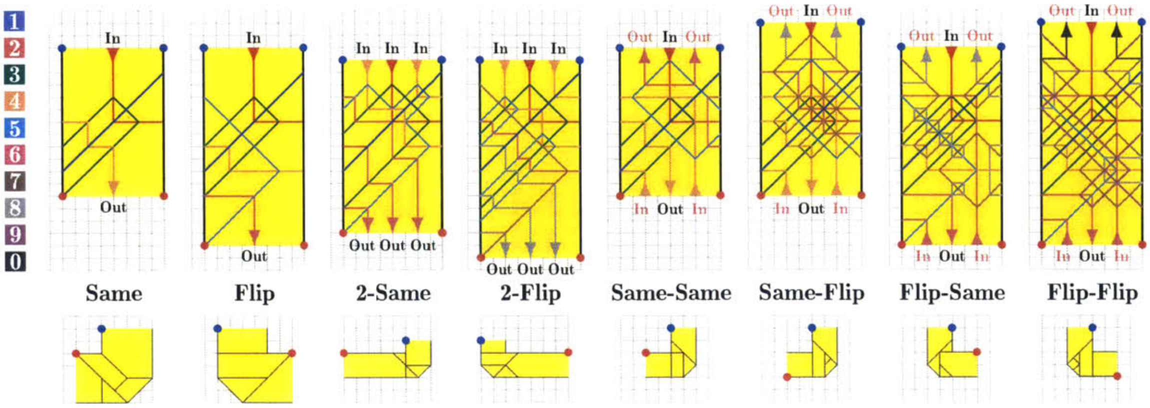


Figure 2-14: Unassigned turn gadgets. Creases must be folded according to color order on left. Input and output creases are labeled with arrow heads, forward signals in black and return signals in red.

Theorem 2.2.5. *The unassigned simple-foldability decision problem for square paper with paper-aligned creases at multiples of* 45° ⊠ *is strongly NP-complete in the some-layers and all-layers models.*

Proof. We will use the same techniques from the proof of Theorem 2.2.3 to build an approximation (small corners missing) of an orthogonal polygon from a square, propagating a signal along the paper to force construction parts of the orthogonal polygon, and then invoke the proof of Theorem 2.2.4. However, since both the Arm and the Cage are necessary for the arguments of the latter proof, we will need to enforce construction of the entire orthogonal polygon before Wrapper creases can execute, not just the Cage. We force the entire orthogonal polygon to be constructed, first by propagating a signal throughout the length of the polygon, and then back to the Wrapper using the eight turn gadgets shown in Figure 2-14.

Just as for the assigned gadgets in Figure 2-11, the relevant creases in each gadget have a fixed order that ensure the output crease(s) of a signal may only fold if the input crease(s) have already been folded. When chained together, these signals enforce the order in which turns are constructed and completed.

We split the gadgets into three groups: Simple turns (Same, Flip), Double turns

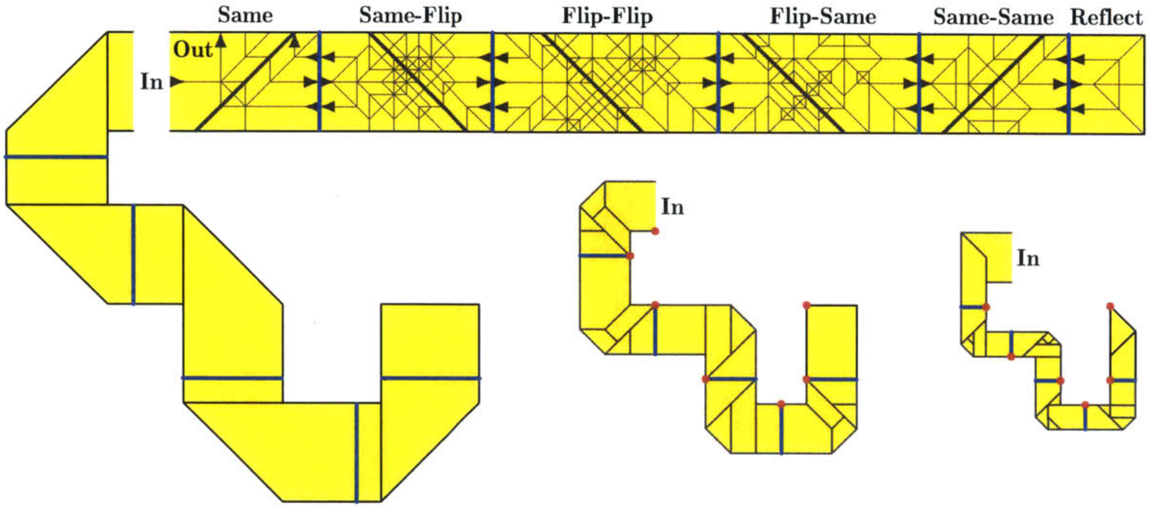


Figure 2-15: Example collection of turn gadgets connected in series demonstrating forward and return signal propagation.

(2-Same, 2-Flip) and 2-Way turns (Same-Same, Same-Flip, Flip-Same, Flip-Flip). Simple and Double turns encode only a forward signal and are completed once the output has been folded, the only difference being that the Double turns are folded in half twice. Alternatively 2-Way turns encode both forward and return signals, respective outputs only foldable if respective inputs have been folded, the return signal folding after the forward signal. The forward signal is propagated along the center axial crease as in the Same gadgets, while the return signal is propagated on the sides. The naming of the 2-Way gadgets are analogous to the Simple gadgets: Same-Flip meaning the original edge of the long rectangle is on the same side as the turn when propagating the signal forward, with the original edge opposite the turn upon the return. An example assembling many of these gadgets coupled in a series is shown in Figure 2-15.

Note that we can trivially connect Double and 2-Way turns together, while Single turns may also interface with them by adding additional folds as shown in Figure 2-15. Figure 2-15 also depicts a Reflection gadget that turns the forward input signal around, propagating from the center to the return outputs on the outside. In this example, the only creases foldable using simple folds from the start are the set of diagonal creases C shown as bold lines in the crease pattern, the folded result shown in the left diagram. Note that these creases don't all have to be folded at the same time, but each must be folded before a forward signal may past through them. Because of the inputs and outputs of each gadget are chained, the first simple fold not in C that may fold contains the crease labeled "In", which when folded will unlock a series of simple folds to propagate the forward signal to the reflect gadget as shown in the middle diagram. The return signal folds may then be executed, ending with the simple fold labeled "Out". The final flat folding is shown in the right diagram.

Now we follow the same construction from the proof of Theorem 2.2.3, constructing an appropriately long rectangle of width four units to be shaped into an approximation of the orthogonal polygon in Figure 2-10, with Wrapper modified to be two units high as in Figure 2-13. We begin construction from the tip of the Arm using Double turns all the way to the top of the Staircase, allowing appropriate space between gadgets so that the constructed polygon has the correct dimensions. The paper will not overlap where creases are folded because the constructed polygon is always three units away from the rest of the polygon already constructed. The Wrapper can be constructed double the width by switching to Single turns on the ends. The construction proceeds to fold the rest of the cage using 2-Way gadgets with a reflect gadget at the end, with the return signal ending by folding the right edge of the wrapper.

The crease pattern resulting from this construction can only fold in the order enforced by the chain of connected gadgets, by the analysis of the gadgets above. Recall that the first crease to fold of the modified Wrapper from Figure 2-13 is a diagonal crease terminating on the right edge of the wrapper which will reflect across the last crease of the return signal, and won't be foldable unless the entire orthogonal polygon approximation has been constructed. Then the same argument as the proof of Theorem 2.2.4 proves the claim directly. □

2.2.7 Infinite, Orthogonal Paper/Orthogonal Creases

In the infinite one-layer or some-layers models, a simple fold must fold one (or more) layer(s) everywhere in the intersection of the fold axis and the valid flat folding. This

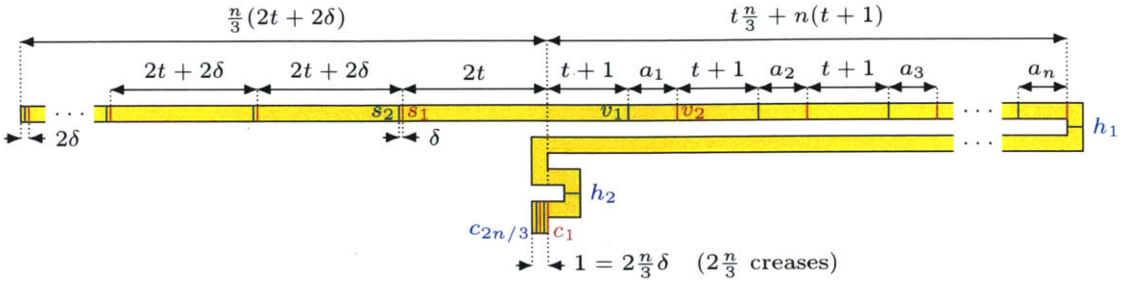


Figure 2-16: An orthogonal simple polygon with mountain-valley assigned paper-aligned orthogonal creases (drawn in red and blue respectively) constructed from an instance of 3-PARTITION that can be folded in the infinite one-layer model if and only if the instance of 3-PARTITION has a solution.

is more restrictive than the one-layer model as foldability in the infinite one-layer model implies foldability in the one-layer model but not the reverse.

Theorem 2.2.6. *The assigned simple-foldability decision problem for orthogonal paper with paper-aligned orthogonal creases* ⊞ *is strongly NP-complete in the infinite one-layer and infinite some-layers models.*

Proof. The proof is again a reduction from 3-PARTITION. Given an instance of 3-PARTITION with integers $A = \{a_1, \dots, a_n\}$ to be partitioned into $n/3$ triples each with sum $\sum A/(n/3) = t$, construct an orthogonal polygonal paper P with paper-aligned orthogonal creases Σ and assignment $\alpha : \Sigma \to \{M, V\}$ as shown in Figure 2-16, with width one everywhere. For our construction we assume each a_i is sufficiently close to $t/3$: if not, add a large number to each so that they are.

There are three functional sections of the polygon. The paper above crease h_1, called the *Pleater*, encodes the integers on the right, and the sets to be satisfied on the left using a pair of creases for each. The paper between creases h_1 and h_2, called the *Base*, is uncreased paper used to exploit the one-layer infinite model. Without loss of generality, we assume the Base remains fixed during folding. The paper below crease h_2, called the *Checker*, can only be completely folded if the input 3-PARTITION instance has a solution. The $2n$ creases on the right of the Pleater encode each a_i

with two vertical creases, one mountain and one valley separated by distance a_i, each pair separated from the others by distance $t+1$. Call this set of creases V containing creases v_i labeled $i \in \{1, \ldots, 2n\}$ increasing from left to right. The $2n/3$ creases on the left of the Pleater come in pairs bounding small distance $\delta = \frac{3}{2n}$, each pair separated from each other by $2t + 2\delta$. Call this set of creases S containing creases s_j labeled $j \in \{1, \ldots, 2n/3\}$ from right to left. Lastly, let C be the set of $2n/3$ creases in the Checker alternating M/V, containing creases c_j labeled $j \in \{1, \ldots, 2n/3\}$ from right to left.

First, if the 3-PARTITION instance has a solution, then (P, Σ, α) is foldable under the infinite one-layer model. Fold explicitly using the following procedure. First fold the two horizontal creases h_1 and h_2. Then choose a triple of a_is in the 3-PARTITION solution and pleating their corresponding creases v_{2i-1} and v_{2i}. These three pairs are foldable under the infinite one-layer model by folding first v_{2i} then v_{2i-1} for each a_i in the triple. Pleating all creases corresponding to a valid triple moves the creases in S to the right by exactly $2t$, aligning s_1 and s_2 with c_1 and c_2 respectively. Now aligned, these creases then be pleated together, moving creases c_3, c_4, s_3, and s_4 to the locations where c_1, c_2, s_1, and s_2 used to be respectively, serving as an invariant. Repeating this process $n/3 - 1$ more times successfully folds all creases.

Second, if (P, Σ, f) is foldable under the infinite one-layer model, then there exists a solution to the 3-PARTITION instance. We first prove two intermediate results: (1) each crease in C can only fold if aligned and folded with some crease in S; and (2) creases s_1 and s_2 must be the first and second creases in S to fold, and must fold aligned with creases c_1 and c_2 respectively.

Proof of (1). By construction, the infinite line induced by each crease c_i will always overlap some part of the Base (which contains no creases) for any folded or partially folded configuration. Thus in order to fold c_i, some other crease must align with c_i on top of the Base. Clearly c_i cannot align with any crease in V or any other crease in C, so it must align with some crease in S. So for any valid folding, there exists a bijection between creases in C and S.

Proof of (2). Suppose for contradiction $s_i \neq s_1$ is the first crease in S to be folded.

Then one of two cases apply. Either s_i folds without aligning with some crease in C, a contradiction by (1); or s_i is folded aligned with some crease in C by folding some subset of creases $V' \subset V$, with s_1 not yet folded and strictly to the right of all creases in C. But since the creases in V' cannot be unfolded, the distance between s_1 and any crease in C can only increase further, and s_1 will never align with a crease in C, a contradiction.

Further, since the horizontal position of s_1 is purely a function of the folded state of creases V and only integral distances exist between folds in V, the horizontal position of s_1 can only change by integral amounts. The only crease of C that is an integral horizontal distance from s_1 is c_1, so they must fold together. Additionally after s_1 and c_1 are folded, s_2 and c_2 are also aligned and must be the next creases to be folded. Suppose for contradiction they were not. We cannot fold any other crease in C or S since no other pair are aligned with each other; and folding some crease in V prevents s_2 from ever aligning with a crease in C, a contradiction.

Now we prove the claim. By (1), creases s_1 and s_2 fold before all other creases in S, aligned with creases c_1 and c_2 respectively. In order to align these creases, some subset of V must have been folded to shift s_1 to the right by exactly $2t$. With s_1 and s_2 so aligned, no $t + 1$ section between a_i sections can be flipped from their original orientation or else s_1 would have shifted to the right by more than $2t$. Further, since a_is are close in value to $t/3$, exactly three a_is that sum to t must have been flipped, i.e. v_{2i-1} and v_{2i} must have been folded from some triple of a_is that sum to t.

Once s_1 and s_2 have been folded, the paper now represents a smaller instance of 3-PARTITION with three fewer a_is that sum to t with identical structure. The remaining creases of S have shifted to the right by $2t + 2\delta$ and the remaining creases of C have shifted to the right by 2δ; in particular, s_3, s_4, c_3, and c_4 are in exactly the same horizontal locations respectively that s_1, s_2, c_1, and c_2 used to be. (2) continues to apply recursively, constraining the next crease pair to fold only after new a_i triples summing to t have been identified and folded. Thus, if (P, Σ, f) is foldable in the infinite one-layer model, there exists a solution to the 3-PARTITION instance.

The theorem follows directly. The reduction is polynomial since the construction is bounded by a $4tn/3 \times 8$ rectangle with $2n + 4n/3$ creases. Further, solutions can be checked naively in $O(n^2)$ time by performing each simple fold in order while checking for self intersection after each fold. □

This reduction only applies in the infinite one-layer model; in the one-layer model, the constructed crease pattern folds trivially. Surprisingly, none of the above arguments relied on knowing the M/V assignment of the creases. For creases C to ever fold, creases h_1 and h_2 must be folded in the same direction; the creases in V must pleat a_i intervals with alternating crease assignment, and the same is true of the creases in S and C. Thus, the theorem also holds in the unassigned case.

Theorem 2.2.7. *The unassigned simple-foldability decision problem for orthogonal paper with paper-aligned orthogonal creases* ⊞ *is strongly NP-complete in the infinite one-layer and infinite some-layers models.*

2.2.8 Infinite, Rectangle Paper/Orthogonal Creases

For assigned crease patterns on rectangular paper with paper-aligned orthogonal creases, Arkin et al. show that determining simple-foldability can be decided in polynomial time in the one-layer, some-layers, and all-layers models, noting that the answer is automatically no in the one-layer model for crease patterns containing both horizontal and vertical creases. Note that when unassigned, all rectangular paper with paper-aligned orthogonal creases (⊞) can be produced by folding the horizontal folds in order alternating mountain and valley, followed by similarly pleating the vertical folds. We prove the same results apply in the infinite one-layer, infinite some-layer, and infinite all-layer models, because the corresponding non-infinite models are equivalent for (⊞) crease patterns.

Theorem 2.2.8. *Concerning simple-foldability of rectangular paper with paper-aligned orthogonal creases* ⊞ *, the infinite (one, some, all)-layer models are equivalent to the (one, some, all)-layer models respectively.*

Proof. The only difference between the infinite and non-infinite versions of simple folds models is that the infinite versions must fold at least one layer everywhere paper exists along the fold axis, while the non-infinite versions do not. Assume for the sake of contradiction that the models are not equivalent so that in a given valid flat folding (P, f, λ) a simple fold may occur that folds paper $U \subset P$ about the fold axis ℓ for which $f(U) \cap \ell$ does not equal $f(P) \cap \ell$. Let q be a point on the boundary of the former but on the interior of the latter which exists since $f(P) \cap \ell$ is line segment. Some p exists in the preimage $f^{-1}(q)$ that is not the endpoint of an already folded crease or else the paper would be discontinuous. Then the crease containing p bounds two facets, of which one facet F intersects U but is not contained in U or else q would not be a boundary point. But rotating $F \cap U$ without rotating $F \setminus U$ would violate isometry, a contradiction. □

We have shown many results known previously only to be weakly NP-complete to be strongly NP-complete, namely the simple-foldability decision problem under the one-layer, some-layers, and all-layers models. Moreover we prove the result holds even for square paper with creases at multiples of 45° in the some-layers and all-layers models; and for orthogonal paper with orthogonal creases in the infinite one-layer and infinite some-layers models. With the formalization of the infinite simple folds models, we also argued their equivalence with respective non-infinite models when folding rectangular paper with orthogonal creases. In doing so, we largely characterize the space of simple folds, though we pose the remainder of the space as open problems for the future.

Chapter 3

Material Thickness

This chapter builds on joint work with Demaine [36] published in the proceedings of the 2015 ASME 39th Mechanisms and Robotics Conference.

While much of the research in computational origami applies to folded surfaces with zero thickness (particularly structures that fold flat), modeling folding surfaces with nonzero thickness is of practical interest for mechanical engineering. Design approaches for folding thick material have many varied applications from kinetic architecture [54] and solar panel deployment [49], to robotics [6] and nano-fabrication [5]. These applications have motivated research into the mathematics and mechanics of rigidly folding thick materials [32, 44, 52]. We discuss some of the existing techniques for taking into account material thickness in the following section.

In this section, we propose a new approach for accommodating thickness that modifies certain existing crease patterns into new planar folding patterns, preserving some structure of the old crease pattern while folding a form whose facets are separated from one another in the final state. We describe a systematic and broadly applicable algorithm to transform an input flat-foldable crease pattern into a new crease pattern having a facet-separated, nearly flat folded state.

Our approach for converting flat foldings into facet-separated foldings replaces each flat crease in the input crease pattern by two parallel creases symmetrically offset about the original at a distance proportional to an assigned crease width satisfying certain properties of the original crease pattern. Instead of one crease folding flat with a turn angle of 180°, the two new creases have a turn angle of 90°. This crease widening creates difficulties at crease-pattern intersections since the offset creases no longer converge to a point. Material in the vicinity around each crease-pattern vertex is thus discarded to accommodate crease widening. While this modification creates holes in the material, it introduces extra degrees of freedom that can allow the widened creases to fold. Additionally the algorithm identifies and removes some surface material on one side of creases to avoid self-intersections.

We provide conditions on input flat folded states for the algorithm to produce a thickened crease pattern avoiding local self intersection, namely that crease-pattern faces are convex and creases do not touch the insides of other creases in the input. We also provide bounds for the maximum thickness that the algorithm can produce for a given input. We demonstrate our results in parameterized numerical simulations and physical models.

3.1 Existing Techniques

There are many existing approaches that seek to account for material thickness in folding applications, each with their own strengths and weaknesses. We discuss the techniques below, which are also illustrated in order in Figure 3-1.

3.1.1 Hinge Shift

The hinge shift strategy shifts hinges out of plane to accommodate material thickness [30]. While readily useful in creating one-dimensional foldings of thick material, this technique is harder to apply to 2D crease-pattern networks. Hinges start out of plane so cannot build on existing design techniques starting from a coplanar folding pattern. In addition full range of folding motion is restricted. A recent approach extends the idea of hinge shifting to higher degree crease pattern vertices, but this method is geometrically restrictive in the angles and thicknesses allowed [11].

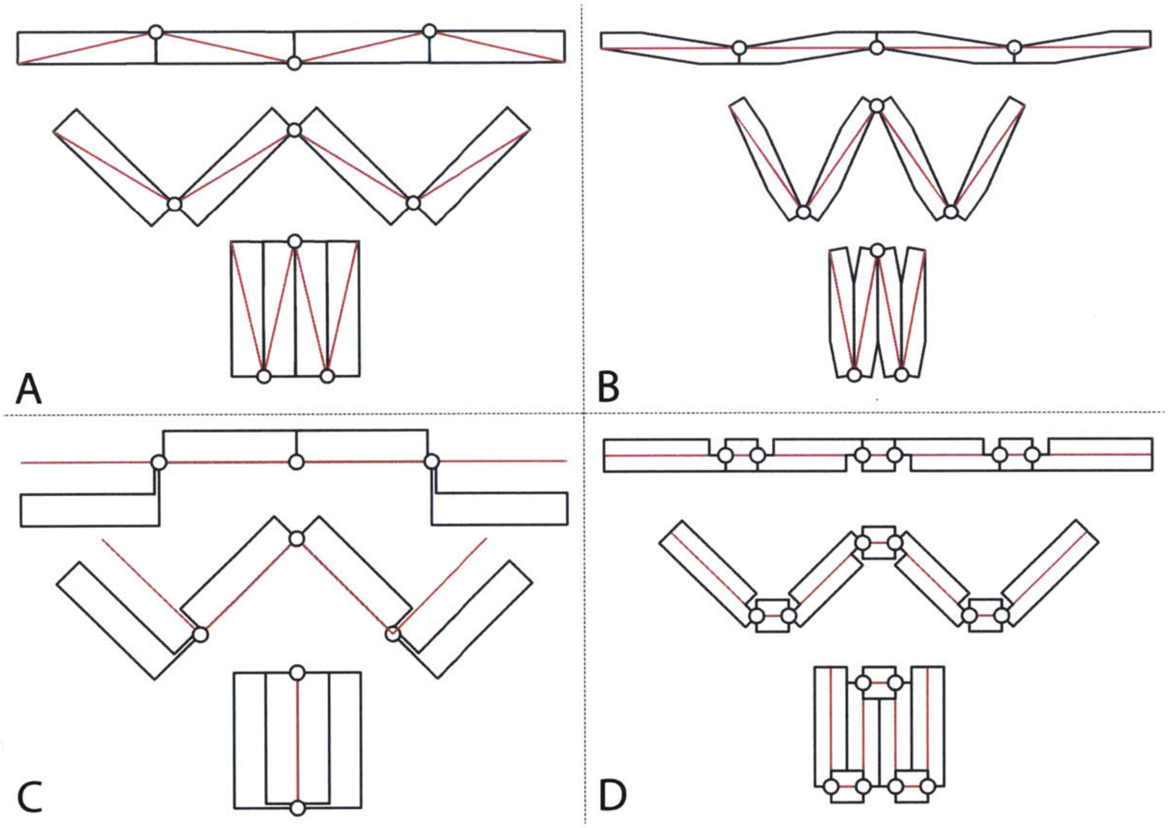


Figure 3-1: Some existing thick folding techniques: (A) Hinge Shift, (B) Volume Trimming, (C) Offset Panel, and (D) Offset Crease.

3.1.2 Volume Trimming

The strategy presented in [54] trims the edges of a thickened surface to overcome many of the difficulties of the hinge shift technique. However, this method also suffers from decreased range of motion and the slanted surfaces can be difficult to fabricate in practice.

3.1.3 Offset Panel

The offset panel technique [23] is probably the most promising in application because it is very flexible while accommodating full range of motion. This method retains hinges at the folding plane but shifts the thick material away from the folding plane. While promising, fabricating such structures can be difficult requiring robust standoffs to connect thick material to hinges.

3.1.4 Offset Crease

In this paper we expand on the ideas presented in [62] which accommodates material thickness by widening creases with flexible material, creating a hinge from a two-dimensional region of material. We propose a modification of the offset crease technique that widens creases in a systematic way, replacing each crease with two ideal hinges without relying on flexible materials. While this technique does not preserve exact structure of the input crease pattern, it creates a structure that can be easier to fabricate than other techniques. Additionally, the proposed technique allows original facets to be parallel in both flat and folded configurations, potentially allowing for alignment of surface mounted components. We describe this technique in detail in the following sections, concentrating first on definitions and then the algorithm itself.

Related to the proposed method are a few other methods for accommodating material thickness. A patent by Hoberman [29] offsets creases in a non-parallel way to accommodate thickness, but also suffers from decrease range of fold angle and does not natural handle crease patterns with internal vertices. Still other methods involve adding degrees of freedom by allowing faces to slide longitudinally along creases, but can be quite difficult to fabricate [58].

3.2 Definitions

We would like to take as input a surface that has been folded flat and output a "thickened" version. In order to perform this task, we must first specify the input precisely, namely the flat folded state. We will describe input flat folded states by way of crease patterns and non-wrapping layer ordering graphs.

Let a *crease pattern* Ξ be a finite straight-line planar graph embedding in $\mathbb{R}^2$. Call crease-pattern edges *boundary edges* if they bound the exterior face, and call them *creases* otherwise. Similarly, call crease-pattern vertices *exterior* if they bound the exterior face with all other vertices *interior.* When we speak of angles around an interior vertex v, we are referring to the cyclically ordered set of angles between adjacent edges connected to v. A crease pattern is said to be *locally flat-foldable* if the alternating sum of angles around every interior vertex is zero. As discussed later, we will also restrict locally flat-foldable crease patterns to have only convex interior faces.

Certainly if we are given as input a flat folded surface, the network of creases on the unfolded surface define a crease pattern which will be locally flat foldable. The next thing to pin down is the ordering of layers in the folded state.

Given a locally flat-foldable crease pattern Ξ, a *flat mapping function* $f_\Xi : \Xi \to \mathbb{R}^2$ is a piecewise isometric mapping under which each interior face of Ξ is congruent, interior faces that share an edge in Ξ share the same edge in $f_\Xi(\Xi)$, and exactly one of any two adjacent interior faces in $f_\Xi(\Xi)$ is reflected from its orientation in Ξ (i.e. each crease has been folded). This function uniquely exists for a locally flat-foldable crease pattern up to isometry (see Figure 3-2).

Here we adapt the work on layer ordering presented in [39]. Given an existing flat folded surface with crease pattern Ξ a *layer ordering graph* Λ is a directed graph on the faces of Ξ with an edge between faces A and B if and only if there exists some points $a \in A$ and $b \in B$ such that $f_\Xi(a) = f_\Xi(b)$ (the faces overlap in the folding).

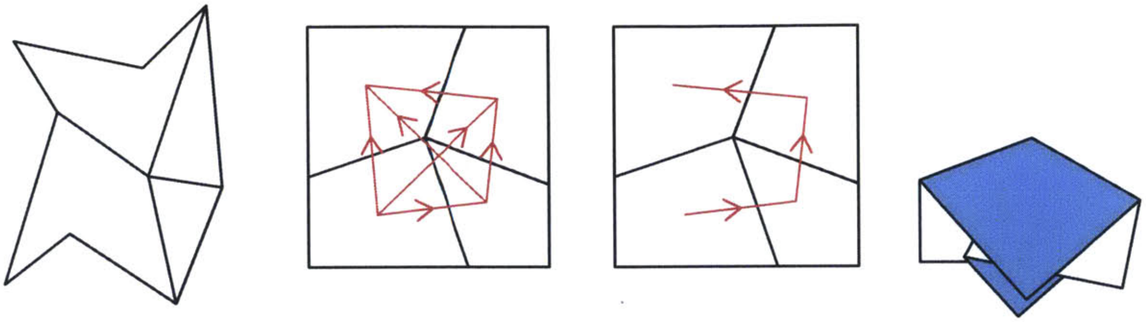


Figure 3-2: From left to right: (1) generic crease pattern Ξ_0, (2) locally flat foldable crease pattern Ξ with layer ordering graph Λ, (3) with reduced layer ordering graph Γ, and (4) flat folding $f_\Xi(\Xi)$.

The direction of the edges in the directed graph are given by arbitrarily calling the surface normal of some face in the flat folding 'up' and drawing edges to point to the face on top of the other. Such a layer ordering may not be well defined if faces are not convex (parts of a face may exist above and below another); as such we will restrict ourselves to crease patterns with convex faces for the remainder of the paper. Additionally, constructing the desired face offset folded state will be impossible if the faces of the layering ordering graph contains a directed cycle because some faces could not be ordered. We will thus restrict to only flat folded surfaces with acyclic layer ordering graphs whose faces can be partially ordered.

Layer ordering graphs can be very complicated, typically containing edges on the order of the squared number of crease-pattern faces. However, they often contain significant redundancy with respect to providing layer ordering information. For example, consider an edge of a layer ordering graph (A, B) from crease-pattern face A to B (B is on top of A), for which there exists some other directed path L from A to B. Transitivity ensures that L enforces the ordering condition imposed by (A, B), so edge (A, B) is redundant and can be removed from the graph without losing any layer ordering information. We then implicitly construct the *reduced layer ordering graph* Γ from the layer ordering graph Λ by identifying any such redundant edge and removing it from the graph. This process terminates and results in a unique output since it is a transitive reduction.

Lastly, we define a flat folded state (Ξ, Γ) as a locally flat-foldable crease pattern together with a reduced layer ordering graph free from self-intersection. Specifically, for any crease ξ bounding faces A and B and a third face C which strictly intersects ξ, no directed path exists in the reduced layer ordering graph Γ between faces A and B visits face C (face C does not intersect crease ξ). This object will serve as the input to our thickening algorithm. Note that a flat-folded state implies a crease assignment to each crease (either mountain or valley) by comparing the orientation and order of faces according to the flat mapping function f_Ξ and Γ. Further, we call the reflex side of a creased surface the *outside* of the crease, and similarly we call the convex side of a creased surface the *inside* of the crease.

A restriction on our approach is if two creases in a crease pattern wrap around each other in the flat folded state, specifically if one crease touches the inside of another crease, self intersection can become a problem. We will go into more detail as we describe the algorithm, but for now we will call an input flat folded state *non-wrapping* if no crease or boundary edge point of the input touches the inside of another crease.

3.3 Algorithm

The goal of this paper is to construct a thickened version of a given a non-wrapping flat folded state (Ξ, Γ). The strategy is to offset crease-pattern faces from their flat folded state consistent with their layer ordering and create new creases to accommodate the offset. First, we must define an offset distance between every pair of faces which implies a width for each crease. Second, we construct scalable polygons at each interior crease-pattern vertex from which material will be removed to accommodate widened creases. Third, we refine the polygons to ensure that each effective vertex does not exhibit local self intersection. Fourth, we calculate a range for allowable scale factors such that vertex polygons do not intersect. Fifth, we lay out the new crease pattern with holes having a non-flat folded state according to a chosen scale in the allowable range. Last, we address constructing the thickness of each face based on avoiding local self intersection. Additional adjustments may then be made to account for global self intersections.

3.3.1 Crease Width

The first goal of the algorithm is to specify a width for each crease in a flat folded state (Ξ, Γ), with all mutually consistent with the layering order of offset faces. Intuitively, we want to separate the layers of the input by nonzero amounts and assign a crease width based on the distance between adjacent faces. If crease widths are chosen small, we can think of the desired output as an "almost flat" version of the original that allows for nonzero space between layers. The concept of crease width is related to the same term applied to the one-dimensional stamp folding problem [59], but we apply it to 2D flat-foldable crease patterns with sortable layer orderings. For our purposes, given reduced layer ordering graph Γ it suffices to choose a positive weight for each directed edge such that given any two interior crease-pattern faces A and B, every path from A to B in Γ has the same weight sum. We will call such a weight assignment $\omega : \xi \in \Xi \to \mathbb{R}^+$.

Such a weight assignment always exists; particularly one can be constructed by choosing an arbitrary linearization of the partial order prescribed by Γ to create a total order, and defining the weight along a crease to be the absolute difference between the layer ordering numbers of the crease's incident faces. By giving a weight to each crease of Γ, we can calculate a crease width for every crease of Ξ by summing the total weight along any path from one face incident to the crease, to the other.

The choice of ω can be viewed as a design choice for the algorithm implementer. One might strive to choose an ω that optimizes some natural metric such as minimizing the maximum thickness of any crease, but the work in [59] and [17] seem to suggest such questions may be NP-hard even for one-dimensional graphs. As such, we do not attempt to optimize the choice of ω here, and leave the exploration in this area as an open problem.

Once we have assigned a crease width to each crease, the construction involves replacing each crease in the input crease pattern with two parallel creases symmetrically offset about the original, separated at a distance proportional to the assigned

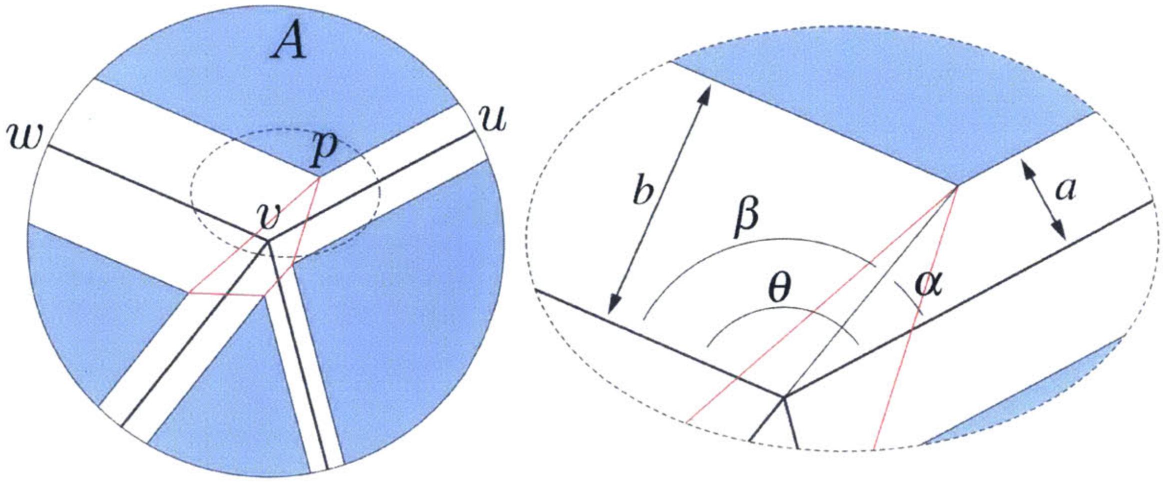


Figure 3-3: Polygon construction. A generic internal crease pattern vertex showing relationship between offsets and angles.

crease width. This replacement creates difficulties at crease intersections since the offset creases will no longer converge to a point. Material in the vicinity around each crease-pattern vertex will need to be discarded to accommodate the widened creases. Next, we will discuss the construction of the region to be discarded.

3.3.2 Polygon Construction

Now that crease widths have been defined, we must interface widened creases with each other in the vicinity of crease-pattern vertices. For each vertex, we construct a polygon that will interface with widened crease lines around the vertex. These polygons will be scalable based on how thick we would like to make the material with respect to the crease pattern, up to a point. We will deal with the allowable range of scaling factor later. First, we must define the geometry of these vertex polygons so they will align with all the crease widths around the vertex.

We want a vertex polygon to contain one vertex per face adjacent to the crease-pattern vertex at a distance from each adjacent crease proportional to the crease width of the crease. Consider crease-pattern vertex v with face A adjacent to it, bounded by adjacent creases $\{u, v\}$ and $\{v, w\}$ with crease widths $2a$ and $2b$ respectively. Let the angle between these creases be θ. Then the location of the polygon vertex p in

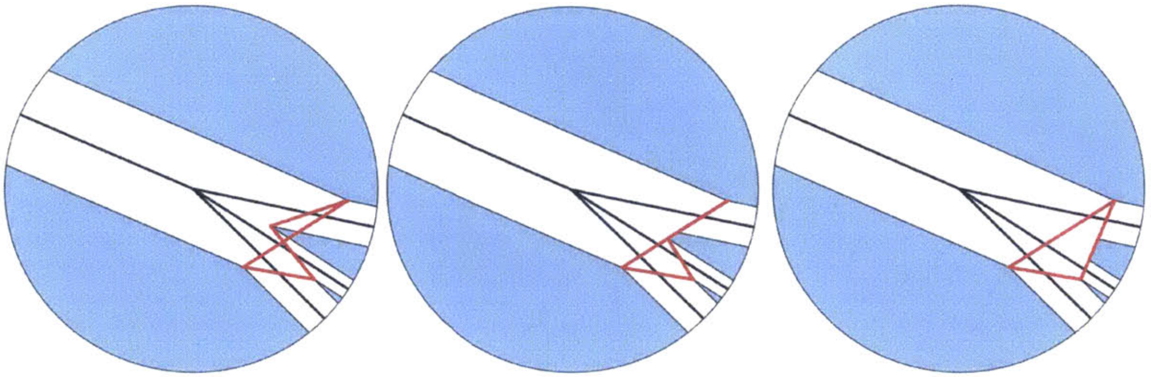


Figure 3-4: A non-simple vertex polygon and refinement by clipping crossings.

this face must be a distance a from crease $\{u, v\}$ and distance b from crease $\{v, w\}$. This point is uniquely defined and can be parameterized by the length h of segment $\{v, p\}$ and the angles α and β between this segment and creases $\{u, v\}$ and $\{v, w\}$ respectively (see Figure 3-3). Some trigonometry reveals that these angles are given by

$$\tan \alpha = \frac{\sin \theta}{b/a + \cos \theta}, \qquad \tan \beta = \frac{\sin \theta}{a/b + \cos \theta} \tag{3.1}$$

with domains $\alpha, \beta \in [0, \pi]$, and $h = a/\sin \alpha = b/\sin \beta$. Repeating this procedure for each face adjacent to an interior crease-pattern vertex constructs points that when connected based on facet adjacency form a polygon. For exterior crease-pattern vertices, the same construction applies except we include the original vertex and intersections between crease width lines and boundary edges in our polygons. We call the regions in each face bounded by offset creases *reduced faces* (shown in blue in the figures). Unfortunately, edges of a constructed vertex polygon may properly cross as in Figure 3-4. However, we can easily modify the vertex polygon to be weakly simple, or even convex, by clipping any facet sector crossing the polygon. More specifically taking the convex hull of the vertex polygon, mark each vertex whose adjacent reduced face does not properly intersect the convex hull. Trimming the intersecting reduced faces against the convex hull of the marked vertices results in an

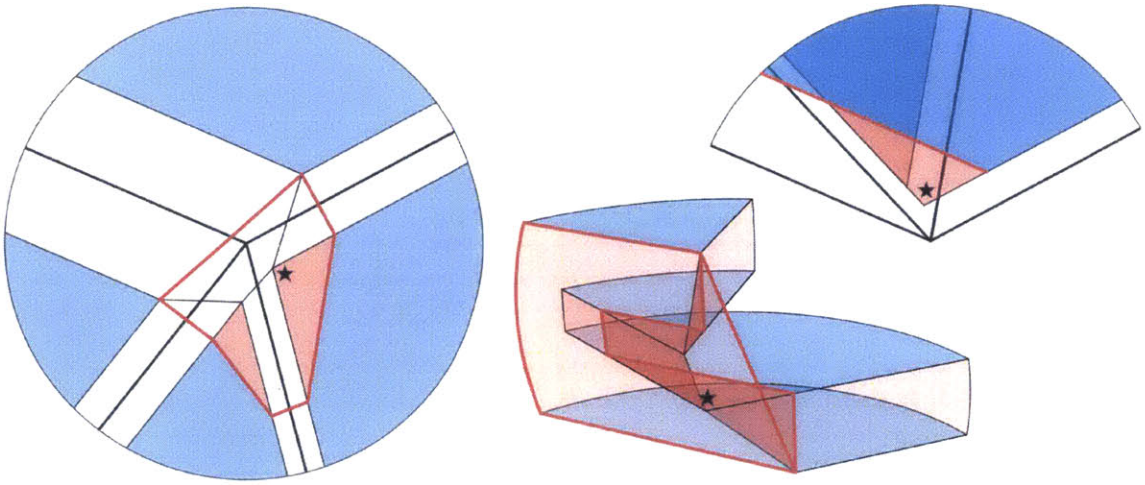


Figure 3-5: Trimming intersecting region.

appropriate convex vertex polygon, though in some cases it may suffice to remove less material (see the middle diagram in Figure 3-4). Note that crossings can only arise if two adjacent vertex angles sum to more than 180°. The reduced faces of these two angles cannot properly intersect the convex hull of the vertex polygon, so at least two marked vertices exist.

Locally, this polygon divides the area around the vertex into three region types: the polygon, widened creases, and reduced faces (the cardinality of the latter two equaling the number of creases adjacent to the crease-pattern vertex). We will use this terminology to talk about these regions in the following sections.

3.3.3 Refinement

The newly constructed creases and polygons in the previous sections serve to locally satisfy isometry between offset faces by removing material at a vertex and adding new creases to accommodate the offset. However, creases with larger crease width require more paper to be absorbed into widened crease regions, reducing the size of surrounding reduced facets. The interaction of this tradeoff between different regions creates the potential for intersection between widened creases and reduced facets. We fix this type of self intersection by checking each widened crease/reduced facet pair for intersection. If they intersect, trim the reduced facet along the widened crease

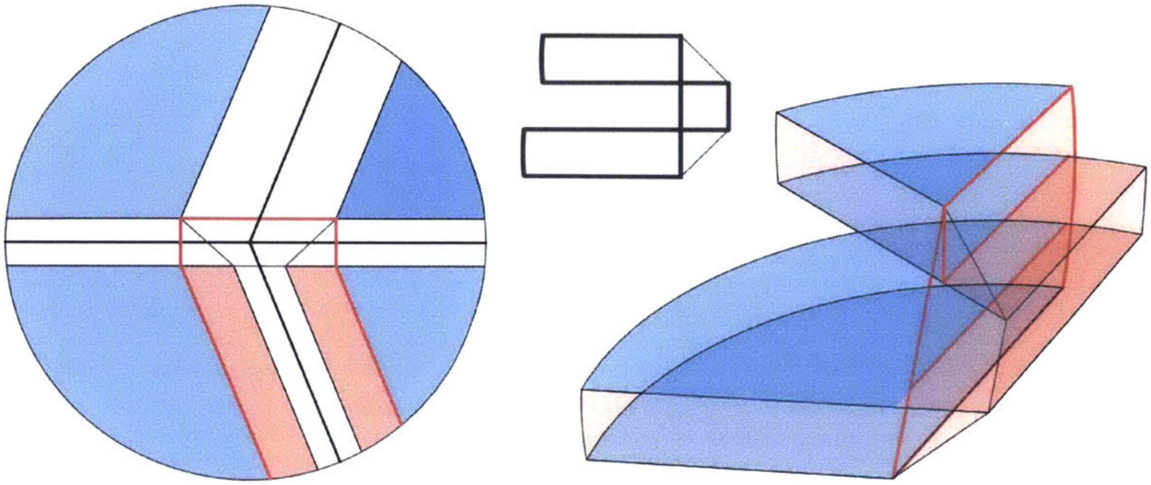


Figure 3-6: Unbounded intersection for inside touching creases in input flat folded state.

boundary and refine the vertex polygon to reflect this change (see Figure 3-5).

There is a worry that this procedure could remove material that is not a bounded distance from the vertex. For example, the crease pattern shown in Figure 3-6 contains two creases that when widened have an intersection that extends to infinity. Fortunately, this type of situation only occurs locally when some crease of the input touches the inside of another crease, which we have forbidden by requiring a non-wrapping input. Reduced facets can only be trimmed a finite number of times because trimming cannot increase the number of intersections, thus the refinement terminates.

3.3.4 Scale Factor

After creating vertex polygons and local widened crease/reduced facet regions that locally do not self intersect, we can determine how large these polygons can be before intersecting each other. Each widened crease edge is bounded on either side by a vertex polygon. Consider crease ξ with length is d. Then each widened crease edge of ξ is shorter than d according to the size of each incident vertex polygon. Let (h_a, α) and (h_b, β) define the locations of the vertex polygon vertices on either side of ξ contained in the same face F. If we let the size of all vertex polygons scale by

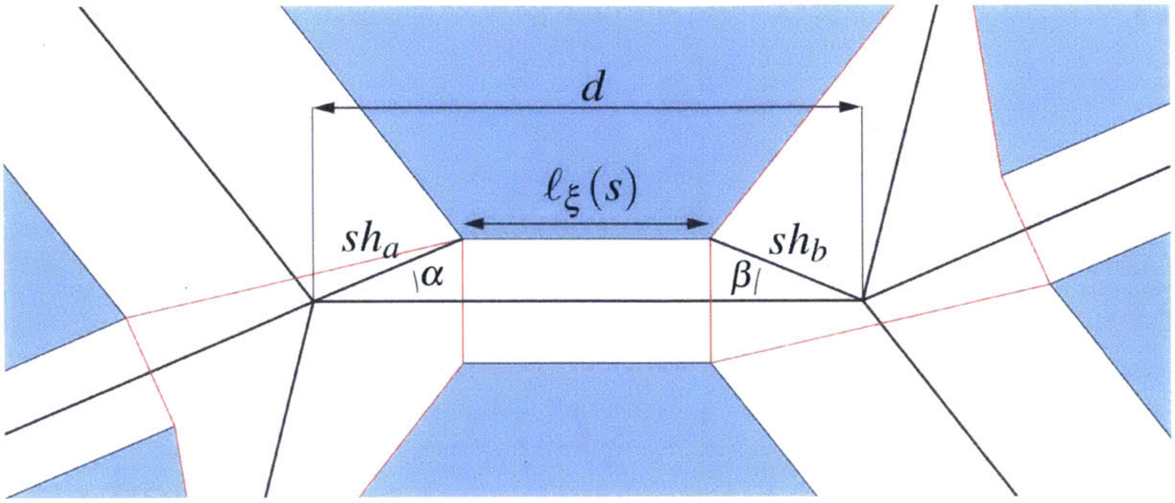


Figure 3-7: Scale factor calculation showing relevant quantities.

a factor s, then the length ℓ_ξ of the widened crease segment in F is given by the following function of s (see Figure 3-7):

$$\ell_\xi(s) = d - s(h_a \cos\alpha + h_b \cos\beta). \tag{3.2}$$

For $(h_a \cos\alpha + h_b \cos\beta)$ negative, $\ell_\xi(s) > 0$ for all $s > 0$ so this crease ξ does not restrict scale. For $(h_a \cos\alpha + h_b \cos\beta)$ positive, there exists some s_ξ strictly positive for which $\ell_\xi(s_\xi) = 0$. This event corresponds to neighboring vertex polygons intersecting which we would like to forbid. Taking the minimum s_ξ over all creases $\xi \in \Xi$ yields a strictly positive upper bound s^* on scale factors by which vertex polygons can be scaled without overlap. Note that for $s = 0$, the crease pattern is not offset at all and facets remain coplanar, and the folded form cannot be produced with material of any finite thickness. Strictly positive s, such as s^* calculated above, allow the modified pattern to accommodate some finite thickness, with a larger s accommodating a larger thickness relative to the geometry of the input crease pattern. Of course this scale s^* only insures that vertex polygons do not interact. It is possible that with this calculated scale, global intersection between faces of the folding can still arise. Nonetheless, we show the following:

Theorem 3.3.1. *Given a non-wrapping flat folded state* (Ξ, Γ) *and weight assignment* $\omega : \xi \in \Xi \to \mathbb{R}^+$*, there exist some positive non-zero scale* s *for which the above con-*

struction globally contains no strict intersection between faces in the three-dimensional folded state.

Proof. Suppose for contradiction that the construction produces intersecting faces for every positive non-zero scale s. Intersection cannot occur between reduced polygon faces because they are offset from each other in a way consistent with the input non-wrapping reduced layer ordering graph containing no self-intersection. Thus, any face-face intersection must exist between a widened crease face and some other face. Let ξ be the original crease corresponding to some widened crease face strictly interesting another face F. Because the input is non-wrapping containing no self-intersection, ξ does not intersect F in the input non-wrapping flat folded state. Increasing the scale s from 0 and performing the above construction results in a continuous parameterized family of three-dimensional foldings. More importantly, let $d(s)$ be the minimum distance between the widened crease associated with crease ξ in a construction with scale s and the reduced polygon formed from face F. Then $d(s)$ is positive for $s = 0$ and varies continuously and weakly monotonically with s. Thus there exists some positive non-zero scale $s' \in (0, s^*)$ for which $d(s')$ is also positive. But this is true for every intersection involving a crease, so there must exist some scale where no intersection occur, a contradiction.

□

3.3.5 Final Construction

Now given a flat folded state (Ξ, Γ) and width assignment ω, we can calculate the upper bound s^* on scale to forbid vertex polygon intersection and choose a scale s' in the range $(0, s^*)$ to construct a modified crease pattern that avoids self intersection. Quite simply the construction is placing vertex polygons scaled by s' and adding widened crease lines parallel to the original creases between vertex polygons. The entire process is shown in Figure 3-8: first the input non-wrapping flat folded state, offset facets, and finally the offset polygons, together with their counterparts in the folding domain.

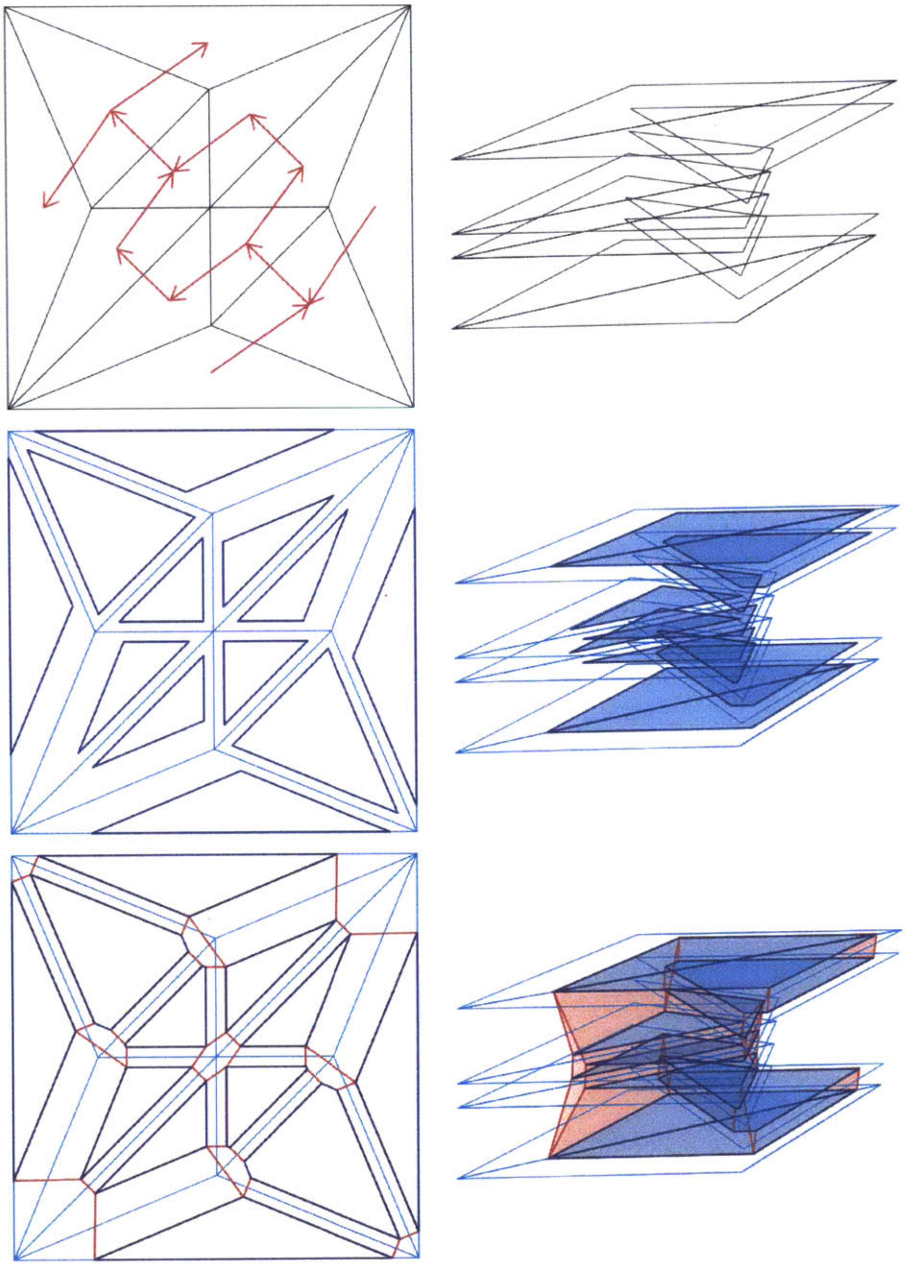


Figure 3-8: Construction process.

Theorem 3.3.2. *Given a non-wrapping flat folded state* (Ξ, Γ) *and weight assignment* $\omega : \xi \in \Xi \to \mathbb{R}^+$, *the construction above terminates in polynomial time.*

Proof. Given the weight assignment, the vertex polygons are each bounded, constructed as described in Step 2 by offsetting the original geometry by finite amounts and connecting vertices, which can be constructed directly in linear time. Clipping ensures the vertex polygons are weakly simple and can be performed naively by comparing each vertex-edge pair in quadratic time. Trimming in Step 3 can also be implemented in quadratic time by checking each pair of faces locally around a vertex. Local intersections of the faces around a vertex are thus removed in the trimming step by construction. Calculating the scale upper bound s^* guaranteeing that vertex polygons do not intersect requires a constant-sized evaluation per edge, while calculating an appropriate s' can be computed by evaluating the appropriate scale for each possible intersection pair in at most quadratic time, and choosing the minimum scale. Thus the procedure can be implemented to terminate in quadratic time which is polynomial. □

3.3.6 Adding Thickness

The above construction creates a modified thin crease pattern that separates faces in the folded form to make room for thick panels. Adding material to the constructed thin surface is relatively easy. In general, if crease widths are chosen arbitrarily, facets can be assigned a range of thicknesses to either side that can be accommodated by the crease widths. However, a simpler and more practical assignment might be to assign the same max thickness to the entire crease pattern as many manufacturing processes could benefit from this kind of uniformity (nano-fabrication, sheet metal construction, etc.). We can simply define the max panel thickness t_{max} as the smallest crease width assigned to the flat folded state.

However, this panel thickness cannot be added everywhere or material would self-intersect. For example, if finite panel thickness exists everywhere on adjacent faces on the inside valley side of each crease, the crease would not be able to fold at a right angle without the added material intersecting when folded. There are many ways to solve this problem by removing material. We suggest keeping full panel thickness on widened crease regions to strengthen these traditionally weak interfaces, and removing material from the adjacent face incident to the crease. To accommodate widened crease panel thickness on both sides, we must remove a strip of material of width $t_{max}/2$ on either side of the widened crease from the reduced facets adjacent to the crease, only on the crease's inside surface. This modification will ensure that material in the vicinity of creases do not locally self-intersect.

The problem of global material self intersection during a folding motion is a more difficult computational task, though there are existing computational methods for addressing this issue. The offset panel techniques of [23] also point out this problem. We are looking into more efficient techniques to perform global folding motion collision detection to aid real-world design applications.

3.4 Models

3.4.1 Implementation

We wrote a program to implement the algorithm presented for generating modified offset crease patterns from input flat-foldable crease patterns. The program was written in coffeescript and can be found at http://jasonku.scripts.mit.edu/thick. The input is a vertex set and an ordered list of faces. The program then allows the user to adjust the distance between faces by pressing arrow keys, allowing the user to view the how the crease pattern changes in real time. Figure 3-9 shows a screen shot of the implementation.

3.4.2 Simulations

We developed numerical and physical models to demonstrate the algorithm presented above. We used the algorithm described to modify two existing rigid-foldable flat-foldable crease patterns, the traditional bird base and a modified rigid foldable flap-

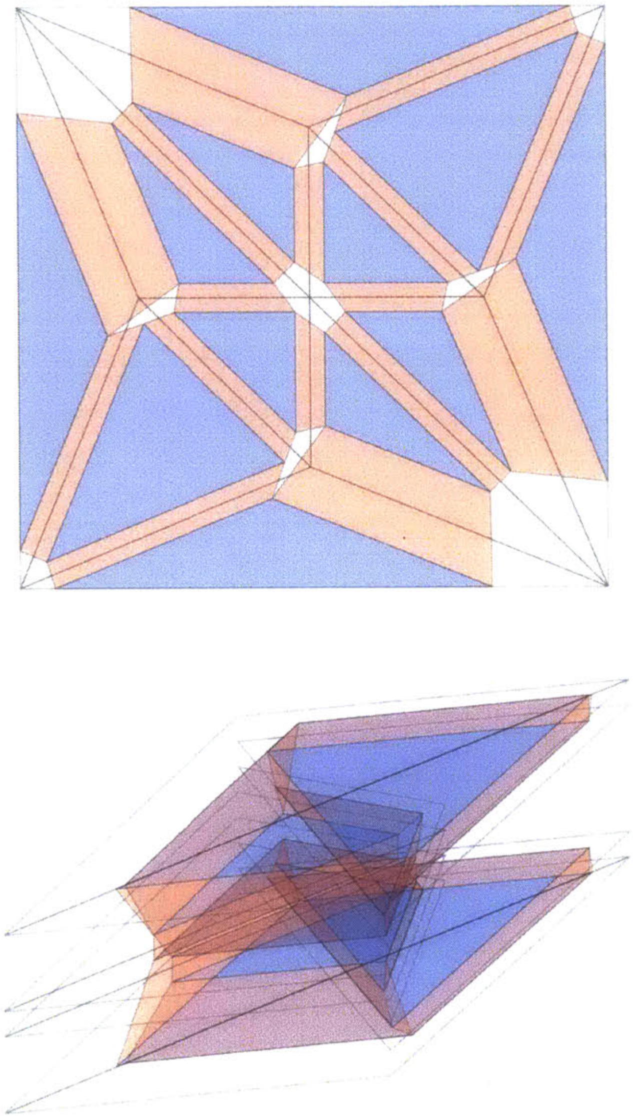


Figure 3-9: A screenshot of our offset crease implementation in action. The model shown is a traditional bird base with uniform thickness offset.

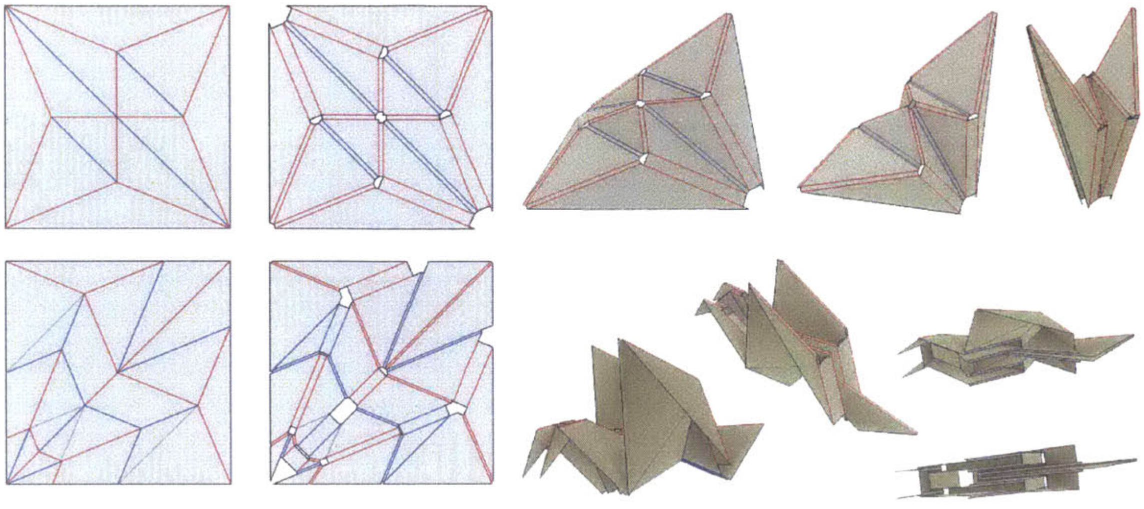
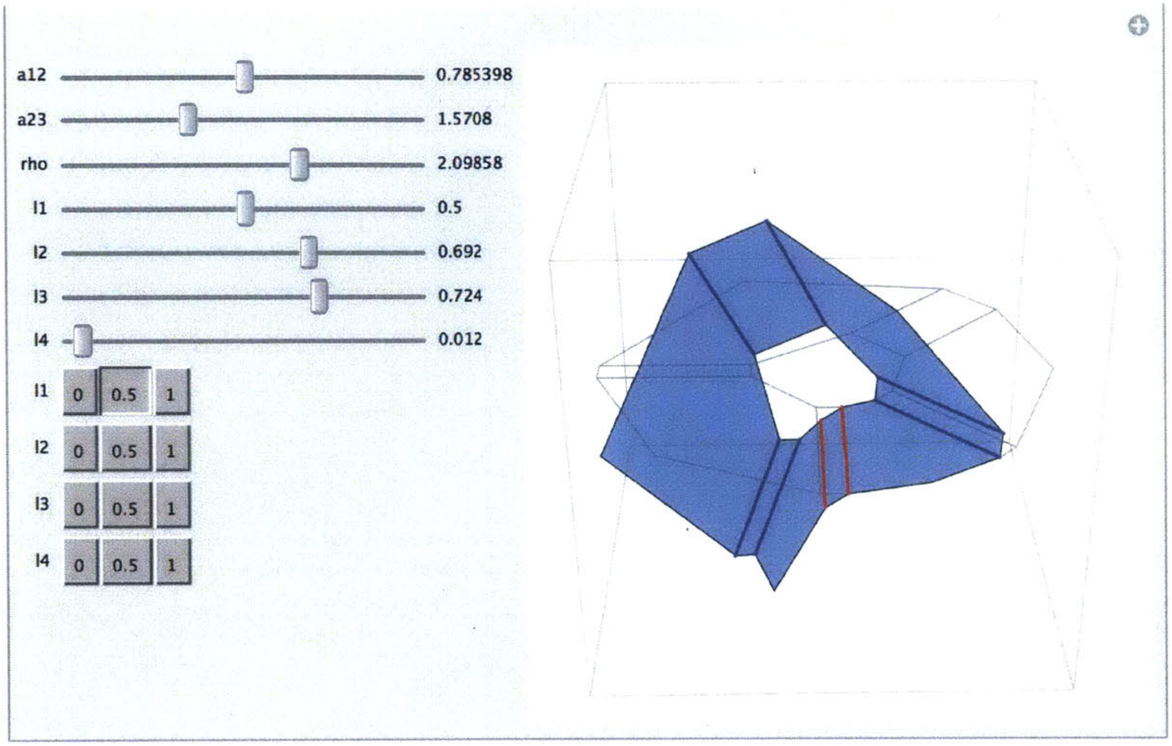


Figure 3-10: Numerical folding simulation of two thickened crease patterns using Freeform Origami.

ping bird designed by Robert Lang as shown in Figure 3-10 (bird base on top, and flapping bird on bottom). On the left the original and modified crease patterns are shown, followed by snapshots of each crease pattern folding. These modified crease patterns were input into a numerical origami simulator called Freeform Origami [55]. This simulator is able to fold a crease pattern incrementally through rigid folding configuration space while seeking to maintain developability and planarity constraints converging iteratively to within double precision. Folding these crease patterns in the simulator demonstrated multiple rigid folded states throughout the folding process to very high accuracy. These simulations provide evidence that a path through the configuration space exist for complex crease patterns between the unfolded and folded states produced by this algorithm. Such a movement seems possible for single-vertex crease patterns because the number of degrees of freedom of the modified structure should in general increase.

We also used a Mathematica model shown in Figure 3-11 to apply the algorithm to single-vertex crease patterns to try and find a path in the folding configuration space between the unfolded and folded states produced by this algorithm. The model allows the user to change the parameters of the system, namely fold angles between creases and splitting ratios between offset crease pairs, in order to satisfy closure. While



a12 0.785398

a23 1.5708

rho 2.09858

l1 0.5

l2 0.692

l3 0.724

l4 0.012

Figure 3-11: Parameterized thick single vertex construction in Mathematica.

we have not found an analytic closure constraint relating fold angles and splitting ratios, we have been able to achieve closure numerically to double precision for a range of inputs. Our results in this area are preliminary, but we have experimental evidence to support that single-vertex crease patterns thickened with this algorithm have a rigid foldable path between unfolded and folded states. We conjecture that the state space for thickened single-vertex crease patterns is a sphere embedded in the multidimensional parameterized space and will leave further discussion in this area to future work.

3.4.3 Physical

Lastly, a physical model of a thickened version of the traditional bird base was fabricated using both acrylic and aluminum on either side of a Tyvek hinge layer. Some views of the physical models can be seen in Figure 3-12 and Figure 3-13. The components of the model were machined using a laser cutter for the acrylic and Tyvek, and a water jet for the aluminum. Hexagonal through holes were machined in the facet

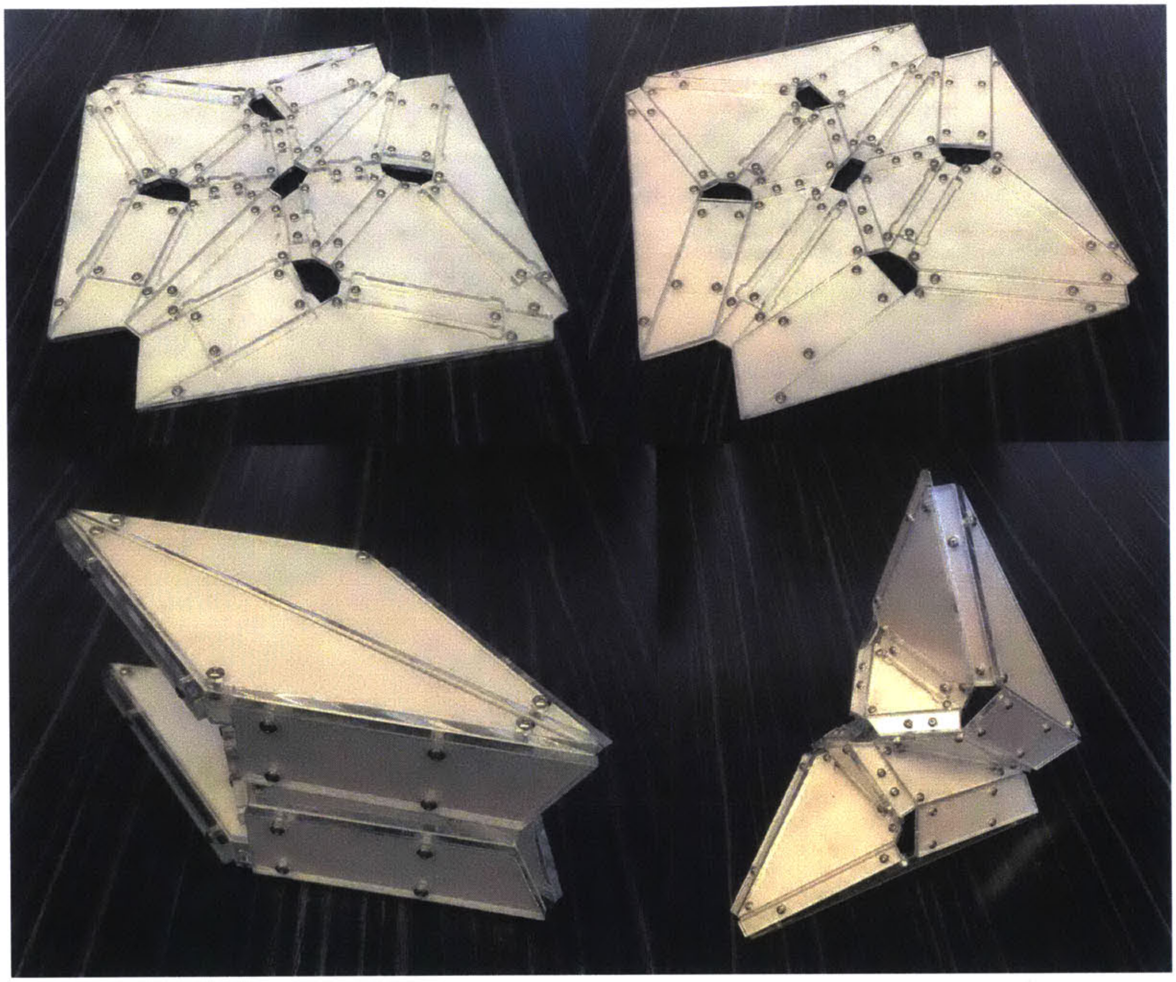


Figure 3-12: An acrylic physical model constructed using the offset crease technique presented.

material into which hexagonal threaded standoffs were inserted. Finally, the layers were assembled and secured together with machine screws.

The folding action observed with this model agrees well with the folded states of numerical simulation, and the motion feels tightly constrained in contrast to the folding mechanisms described in [62]. Empirically fixing the dihedral angle between sector faces while adjusting the angle ratio at one crease, a continuous adjustment of the other crease ratios was observed, also supporting the spherical configuration space conjecture.

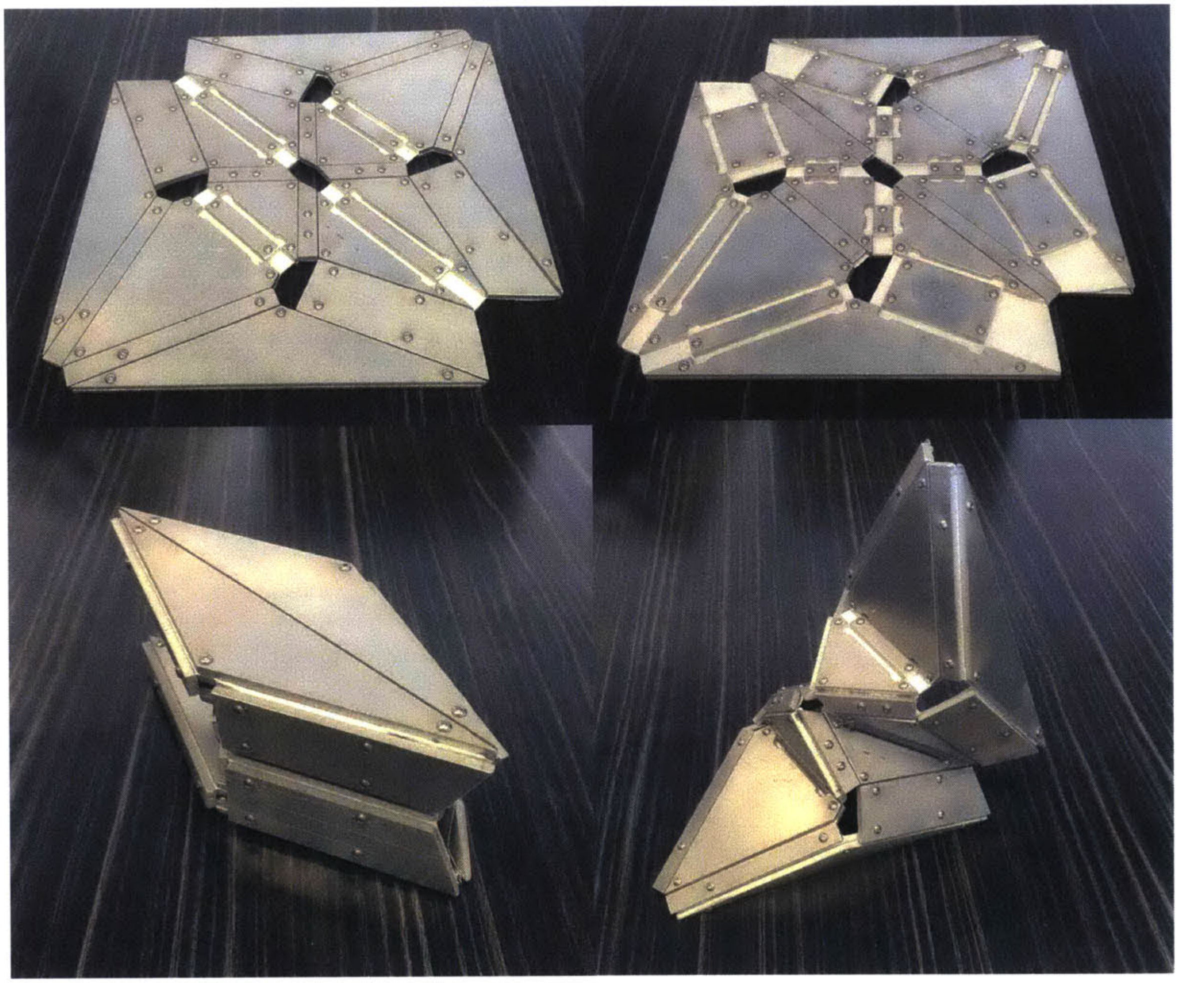


Figure 3-13: An acrylic physical model constructed using the offset crease technique presented.

3.5 Remarks

The algorithm proposed has many benefits over existing thick folding techniques. Facet surfaces in the produced structure's unfolded state are coplanar allowing for ease of fabrication in layer-by-layer manufacturing processes. These same surfaces are parallel in the produced structure's folded state allowing any surface mounted components to mate naturally. Panel thicknesses can be adjusted according to material and scale within bounds provided by the algorithm. Further, every finite area of the algorithm's produced surface may be assigned non-zero thickness, allowing for the production of strong and tightly constrained mechanisms.

The offset crease method provides a thickened folded state suggesting a full range of folding motion as well as parallel facets when fully folded. Assigning crease widths to comply with the acyclic layer ordering of the input flat folded state provides a flexible design space for varied applications, while still constructing one non-trivial folded state with planar facets. While it is still open whether a path of rigid folded states exists through the configuration space in general, there is evidence that one exists for single vertex crease patterns given our numerical models. While compensating for material thickness is not as difficult for non-flat foldings, because faces do not meet each other when folded, we are exploring ways of extending this method for non-flat foldings, particularly those containing face-to-face contact in their non-flat folded form.

Chapter 4

Geometry

This chapter builds on joint work with Demaine [19] published in the Origami^6 collection from the 6th International Meeting on Origami in Science, Mathematics and Education in 2014. Some developments were also presented at the 2016 Joint Mathematics Meetings AMS Special Session on Origami Methods and Applications.

Many problems in origami require the folder to map the perimeter of a piece of paper to some specified folded configuration. In the tree method of origami design, circle packing breaks the paper up into polygonal molecules whose perimeter must be mapped to a specific tree. The fold-and-cut problem inputs a set of polygonal silhouettes whose perimeters must be mapped onto a common line. These two problems are well studied; one solution to the molecule folding problem is the universal molecule [38] while a solution to the fold-and-cut problem lies in the polygon's straight skeleton [16][8]. Both of these problems can be considered as specific versions of a more general problem: the hole problem.

Given a crease pattern with a hole in it (an area of the paper with the creases missing), can we fill in the hole with suitable creases? More precisely, given a sheet of paper and a prescribed folding of its boundary, is there a way to fold the paper's interior without stretching so that the boundary lines up with the prescribed boundary folding? This hole problem was originally proposed by Barry Hayes at 3OSME in 2001 with the motivation of finding flat-foldable gadgets with common interfaces satisfying certain properties, such as not-all-equal clauses for an NP-hardness reduction [9].

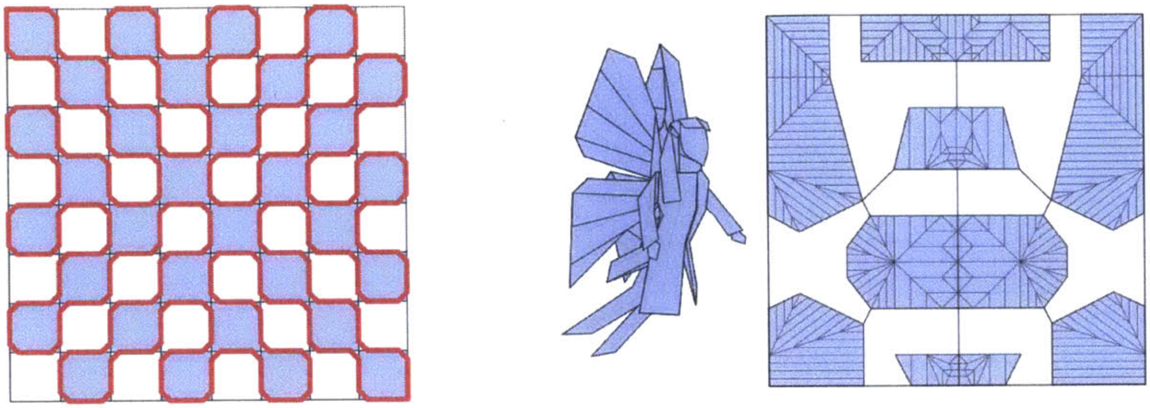


Figure 4-1: (Left) A boundary mapping that might be used to design a color-change checker board model. (Right) An unfinished crease pattern with parts of the crease pattern unknown.

This problem formulation can be transformed to solve several existing problems, as well as some new applications (see Figure 4-1). If we map the boundary to a line, the polygon is now a molecule to be filled with creases or one half of a fold-and-cut problem cutline. The hole problem can also address problems where the boundary is not mapped to a line, i.e. mappings into the plane or into three dimensions, potentially leading to the algorithmic design of multi-axial bases, color-changes, or complex three-dimensional tessellation or modulars. When trying to combine separately designed parts of an origami model, a solution to the hole problem could be used to design an interfacing crease pattern between them.

In this chapter, we show that the hole problem always has a solution for polygonal input boundaries folded at finitely many points under the obvious necessary condition that the input folding is nonexpansive, and present a polynomial-time algorithm to find one. We restrict ourselves to isometry and ignore self-intersection, leaving layer ordering (if possible) as an open problem. Section 4.1 introduces notation and defines the problem. Section 4.1.1 discusses the necessary condition which will turn out to be sufficient. Section 4.1.2 constructs vertex creases satisfying local isometry. Section 4.1.3 propagates the creases. Section 4.1.4 describes partitioning polygons. Section 4.2 describes the algorithm. We continue with a modification of the algorithm that can access all possible isometries satisfying the boundary condition, though as crease patterns exist with super-polynomial complexity with respect to the number of folded boundary points, this extended algorithm cannot in general terminate in polynomial time. Lastly, we analyze some simple hole boundaries with the goal of satisfying more than one boundary condition.

4.1 Definitions

First some notation and definitions. Let $\|\cdot\|$ denote Euclidean distance. Given a set of points $A \subseteq B \subset \mathbb{R}^c, c \in \mathbb{Z}^+$ and mapping $f : B \to \mathbb{R}^d, d \in \mathbb{Z}^+$, we say that A is (*expansive*, *contractive*, *critical*) under f if $\|u - v\| \ (<, >, =) \ \|f(u) - f(v)\|$ for every $u, v \in A$, with (*nonexpansive*, *noncontractive*, *noncritical*) referring to respective negations. Critical is the same as isometric under the Euclidean metric, but because we will use the term "isometry" to refer to isometric maps under the shortest-path metric [20], we use a different term for clarity. We say two line segments *cross* if their intersection is nonempty. We now prove two relations on crossing segments under certain conditions using the above terminology, including a generalization of Lemma 1 from [12].

Lemma 4.1.1. *Consider distinct points $p, q, u, v \in \mathbb{R}^2$ with p, u, v not collinear, line segment (p, q) crossing line segment (u, v), and a mapping $f : \{p, q, u, v\} \to \mathbb{R}^d$. (a) If $\{q, u, v\}$ is critical and $\{p, u, v\}$ is nonexpansive under f, then $\{p, q\}$ is nonexpansive under f. (b) If $\{u, v\}$ is critical, and $\{p, u, v\}, \{q, u, v\}$ are nonexpansive under f, then $\{p, q\}$ is nonexpansive under f; additionally if $\{p, q\}$ is critical under f, then $\{p, q, u, v\}$ is also.*

Proof. (a) Consider the following d-dimensional balls: S_0 centered at $f(q)$ with radius $\|p - q\|$, S_1 centered at $f(u)$ with radius $\|p - u\|$, and S_2 centered at $f(v)$ with radius $\|p - v\|$ (see Figure 4-2). $\{p, u, v\}$ nonexpansive under f implies $f(p) \in S_1 \cap S_2$. $\{q, u, v\}$ critical and (p, q) crossing (u, v) implies $S_1 \cap S_2 \subset S_0$. Because $f(p) \in S_0$, $\{p, q\}$ is nonexpansive under f.

(b) Let $x = u + t(v - u)$ be the intersection of (p, q) and (u, v) and let $x_f = f(u) + t(f(v) - f(u))$. Repeated application of Lemma 4.1.1(a) yields $\|x - i\| \geq \|x_f - f(i)\|$

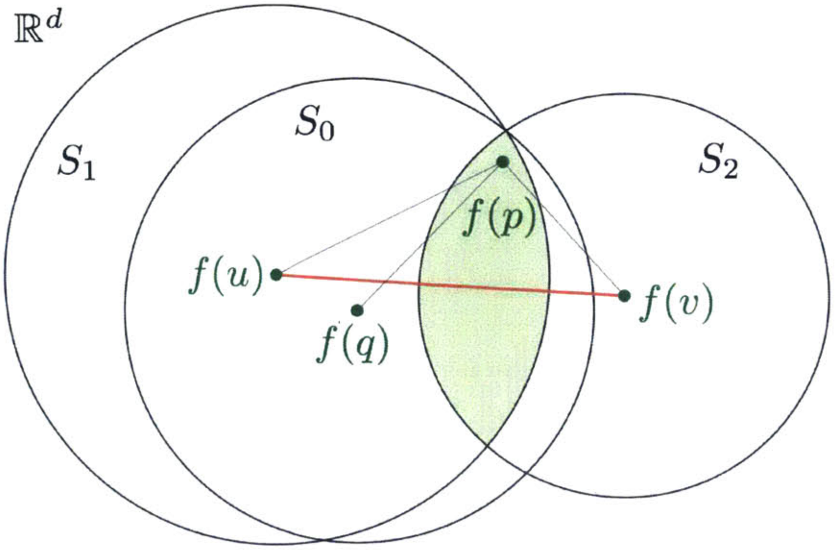


Figure 4-2: Points $f(u), f(v), f(q), f(p)$ with spheres S_0, S_1, S_2. The shaded area $S_1 \cap S_2 \subset S_0$ is the region in which $f(p)$ may exist if $\{p, u, v\}$ is nonexpansive under f.

for $i \in \{p, q\}$. Combining with $\|x - p\| + \|x - q\| = \|p - q\|$ and the triangle inequality, $\|x_f - f(p)\| + \|x_f - f(q)\| \geq \|f(p) - f(q)\|$, yields $\{p, q\}$ nonexpansive under f. Further, if $\{p, q\}$ is critical under f, then so is $\{p, q, x_f\}$. Segments $(f(p), f(q))$ and $(f(u), f(v))$ are coplanar crossing at x_f such that $\{u, p\}$ expansive implies $\{u, q\}$ contractive under f. Since $\{p, q, u, v\}$ is nonexpansive, $\{p, q, u, v\}$ must be critical under f. □

We will consider a *polygon* P to be a bounded closed figure in $\mathbb{R}^2$ bounded by finitely many line segments connected in a simple cycle, with non-touching boundary. This definition restricts polygons to topological disks, and allows adjacent edges to be collinear. Let $V(P)$ denote the vertices of P, ∂P denote the boundary of P, with $V(P) \subset \partial P \subset P$. An edge of P is a line segment in ∂P with endpoints at adjacent vertices. We say that a point $p \in P$ is *visible* from a vertex $v \in V(P)$ if the line segment from p to v is in P. With the terminology in place, we can now state the problem (see Figure 4-3).

Problem 4.1.1. *Given a polygon P in the plane with a boundary mapping $f : \partial P \to \mathbb{R}^d$, find an isometric mapping $g : P \to \mathbb{R}^d$ such that $g(\partial P) = f(\partial P)$.*

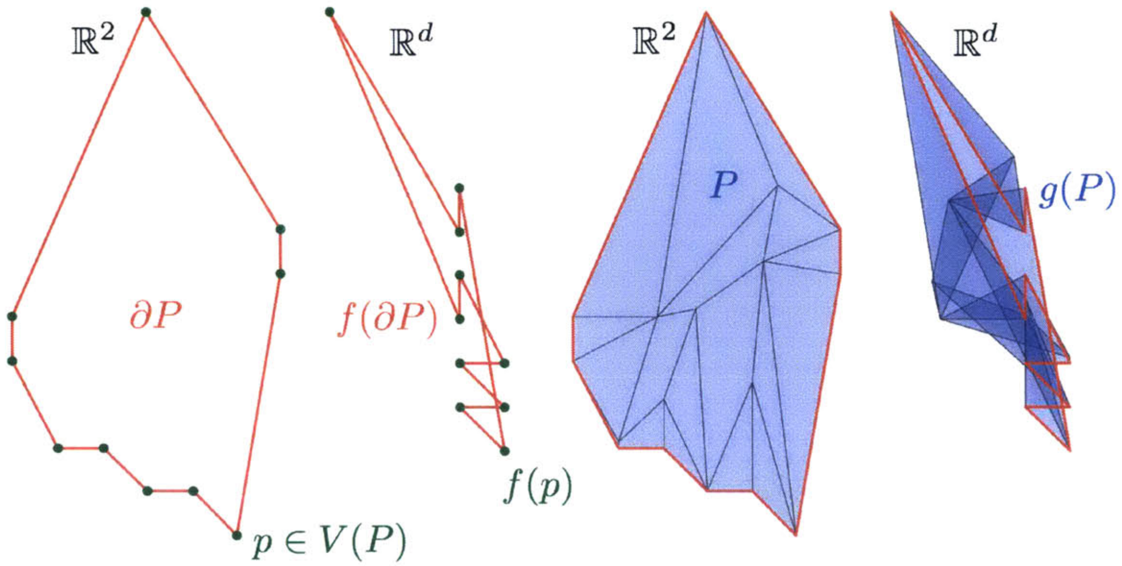


Figure 4-3: Input and output to the hole problem showing notation. Given polygon $P \subset \mathbb{R}^2$ and mapping $f : \partial P \to \mathbb{R}^d$, find isometric $g : P \to \mathbb{R}^d$ such that $g(\partial P) = f(\partial P)$.

If one exists, we call g a *solution* to the hole problem. Mapping P into $\mathbb{R}$ requires infinitely many folds, so we restrict to $d \geq 2$ for the remainder.

4.1.1 Necessary Condition

In this section, we define *valid* boundary mappings and give a necessary condition for the hole problem under the weak assumption that the polygon boundary is folded at finitely many points.

Definition 4.1.1. *$P \to \mathbb{R}^d$, define f to be* valid *if ∂P is nonexpansive under f and adjacent vertices of P are critical under f.*

Lemma 4.1.2. *Consider an instance of the hole problem with input polygon P and boundary mapping $f : \partial P \to \mathbb{R}^d$ nonstraight at finitely many boundary points. If f is not valid then the instance has no solution.*

Proof. Modify $V(P)$ to include boundary points nonstraight under f (vertices adjacent to collinear edges are allowed), so that f is straight for $\partial P \setminus V(P)$. Assume a solution g exists and f is not valid. Then either two points $a, b \in \partial P$ are expansive under f, or two adjacent vertices $u, v \in V(P)$ are noncritical. If the former, then $\{a, b\}$ is also expansive under g, so g cannot be isometric. If the latter, then $f(p)$ is nonstraight for some p on the edge from u to v, a contradiction. □

To determine the validity of f, checking expansiveness between all pairs of points in ∂P is impractical. Instead it suffices to show that the set of vertices is nonexpansive under f, and edges of P map to congruent line segments.

Lemma 4.1.3. *Given polygon P and boundary mapping $f : \partial P \to \mathbb{R}^d$, f is valid if and only if $V(P)$ is nonexpansive and edges of P map to congruent line segments under f.*

Proof. If f is valid, $V(P)$ is nonexpansive under f since $V(P) \subset \partial P$, and edges map to congruent line segments because adjacent vertices are critical and points interior to edges are nonexpansive with endpoints. To prove the other direction, if edges of P map to congruent line segments, adjacent vertices are critical and pairs of points on the same edge are nonexpansive (indeed critical) under f. To show that points from different edges are nonexpansive under f, consider vertex p and point q interior to the edge from vertex u to v. By Lemma 4.1.1(a), $\{q, p\}$ is nonexpansive under f for any vertex p. Now consider point $q' \in \partial P$ not on the edge from u to v. By the same argument as above, $\{q', u, v\}$ is nonexpansive under f, so by Lemma 4.1.1(a), $\{q, q'\}$ is also nonexpansive. □

4.1.2 Bend Lines

When the interior angle of the polygon boundary at a vertex decreases in magnitude under a valid boundary mapping, the local interior of the polygon will need to curve or bend to accommodate. For simplicity, we consider only single-fold solutions to satisfy such vertices, which will still be sufficient to construct a solution. We call these creases *bend lines* made up of *bend points.*

Definition 4.1.2. (Bend Points and Lines) *Given polygon P with valid boundary mapping $f : \partial P \to \mathbb{R}^d$ and vertex $v \in V(P)$ adjacent to two vertices $\{u, w\}$ contractive*

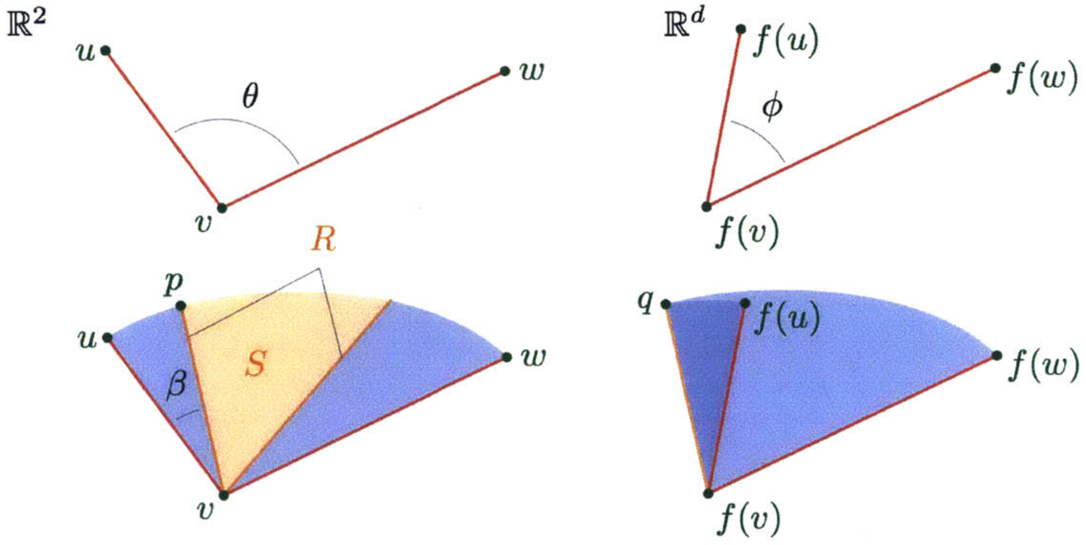


Figure 4-4: The bend points of (P, f, v) showing relavent angles $\{\theta, \phi, \beta\}$, points $\{u, v, w, p, f(u), f(v), f(w), q\}$, and sets $\{R, S\}$. The upper figures show only the boundary mapping, while the lower images show filled, locally satisfying mappings of the interior.

under f, define $p \in P$ to be a bend point *of (P, f, v) if there exists some $q \in \mathbb{R}^d$ (called a* bend point image *of p) for which $\|p - i\| = \|q - f(i)\|$ for $i \in \{u, v, w\}$ and p is visible from v. Further, define a* bend line *of (P, f, v) to be a maximal line segment of bend points of (P, f, v), with one endpoint at v and the other in ∂P; and let a* bend line image *be a set of bend point images of the bend points in a bend line, congruent to the bend line.*

A bend point corresponds to a point in the polygon such that triangles $\triangle pvu$ and $\triangle pvw$ isometrically map to triangles $\triangle qf(v)f(u)$ and $\triangle qf(v)f(w)$ respectively. Bend lines correspond to single folds of P that locally satisfy isometry for the boundary from u to w through v. Lemma 4.1.4 represents bend points explicitly (see Figure 4-4).

Lemma 4.1.4. *Consider polygon P with valid boundary mapping $f : \partial P \to \mathbb{R}^d$ and vertex v adjacent to two vertices $\{u, w\}$ contractive under f. Let $\theta = \angle uvw$ be the internal angle of P at v; let $\phi = \angle f(u)f(v)f(w)$; and let*

$$R = \left\{ p \in P \;\middle|\; \begin{array}{l} \angle pvu \in \left\{\frac{\theta-\phi}{2}, \frac{\theta+\phi}{2}\right\} \\ p \text{ visible from } v \end{array} \right\}, \; S = \left\{ p \in P \;\middle|\; \begin{array}{l} \angle pvu \in \left[\frac{\theta-\phi}{2}, \frac{\theta+\phi}{2}\right] \\ p \text{ visible from } v \end{array} \right\}.$$

Then the set of bend points of (P, f, v) is R if $d = 2$, and S otherwise.

Proof. A point $p \in P$ visible from v is a bend point of (P, f, v) only if triangles $\triangle pvu, \triangle pvw$ are congruent to $\triangle qf(v)f(u), \triangle qf(v)f(w)$ respectively for some bend point image q by definition. Let $\beta = \angle pvu$. If $d = 2$, $\triangle pvu$ and $\triangle pvw$ must be coplanar. Then the internal angles of both triangles at v must sum to θ, and the magnitude of their difference $|(\theta - \beta) - \beta|$ must be ϕ. This condition is satisfied only when $\beta \in \left\{\frac{\theta-\phi}{2}, \frac{\theta+\phi}{2}\right\}$. Thus for $d = 2$, the set of bend points of (P, f, v) is R.

For $d > 2$, triangles $\triangle qf(v)f(u), \triangle qf(v)f(w)$ need not be coplanar. Because $\{u, w\}$ is contractive under f, $\phi \geq |\theta - 2\beta|$, so $\frac{\theta-\phi}{2} \leq \beta \leq \frac{\theta+\phi}{2}$, and points in $P \setminus S$ cannot be bend points. It remains to show that for each point $p \in S$ there exists a satisfying bend point image $q \in \mathbb{R}^d$. For a given p, q must lie on two hyper-cones each with apex v, one symmetric about the segment from $f(v)$ to $f(u)$ with internal half angle β, and the other symmetric about the segment from $f(v)$ to $f(w)$ with internal half angle $\theta - \beta$. These hyper-cones have nonzero intersection H because $(\theta - \beta) + \beta > \phi$ and $\phi \geq \max(\theta - \beta, \beta) - \min(\theta - \beta, \beta)$. The intersection of two hyper-cones with common apex v is a set of rays emanating from v, so H intersects the $(d-1)$-sphere centered at $f(v)$ with radius $\|p - v\|$. Any point in this intersection satisfies all three constraints of a bend point image for any $p \in S$. □

For every $d > 2$, the set of bend points of (P, f, v) is the same, but the set of bend point images increases with dimension. The set of bend point images is a ruled hypersurface of bend line images emanating from $f(v)$. In the case of $d = 2$ above, hyper-cones are simply two rays, leading to disjoint line segments of bend points. For $d = 3$ the set of bend points is a standard cone-like surface. Mapping generally to $\mathbb{R}^d$, the set is a ruled hypersurface of rays emanating from a point.

4.1.3 Split Points

Bend lines locally satisfy the boundary around a vertex with a single crease. We want to find the bend point on a bend line farthest from the vertex that remains nonexpansive with the rest of the boundary. We call such a point a *split point.*

Definition 4.1.3. (Split Points) *Given polygon P with valid boundary mapping $f : \partial P \to \mathbb{R}^d$ and vertex v, contractive under f with every visible nonadjacent vertex, adjacent to two vertices $\{u, w\}$ contractive under f, define p to be a* split point *of (P, f, v), q to be its* split point image, *and x to be its* split end *if*

1. *p is a bend point of (P, f, v), with q its bend point image;*
2. *$\|p - i\| \geq \|q - f(i)\|$ for $i \in V(P)$;*
3. *$\|p - x\| = \|q - f(x)\|$ for some $x \in V(P) \setminus \{u, v, w\}$; and*
4. *p is visible from x.*

Lemma 4.1.5. *Given polygon P with valid boundary mapping $f : \partial P \to \mathbb{R}^d$ and vertex v adjacent to two vertices $\{u, w\}$ contractive under f with v contractive under f with any visible nonadjacent vertex, there exists a split point/image/end triple (p, q, x) for every bend line/image pair (L, L_f) of (P, f, v) with $p \in L$ and $q \in L_f$.*

Proof. Given bend line/image pair (L, L_f) we construct (p, q, x). Parameterize L so that $p(t)$ is the unique point in L such that $\|p(t) - v\| = t$ for $t \in [0, \ell]$ where ℓ is the length of L; and let $q(t)$ be the corresponding bend point image of $p(t)$ in L_f. For any $t \in [0, \ell]$ and vertex x, let $d(t, x) = \|p(t) - x\| - \|q(t) - f(x)\|$. Let t^* be the maximum $t \in (0, \ell]$ for which $d(t, x) \geq 0$ for all $x \in V(P)$, and let X be the set of such vertices $x \in V(P) \setminus \{u, v, w\}$ for which $d(t^*, x) = 0$, and $d(t^* + \delta, i) < 0$ for all $\delta \in (0, \varepsilon]$ for some $\varepsilon > 0$. If we can prove there exists some $x \in X$ from which $p(t^*)$ is visible, then $p = p(t^*)$ is a split point with $q = q(t^*)$ its split point image, satisfying the split point conditions by construction.

Suppose for contradiction that t^* does not exist so that for all $t \in (0, \ell]$, $d(t, x) < 0$ for some $x \in V(P)$. Because d is continuous and $d(0, x) \geq 0$ for all $x \in V(P)$, there

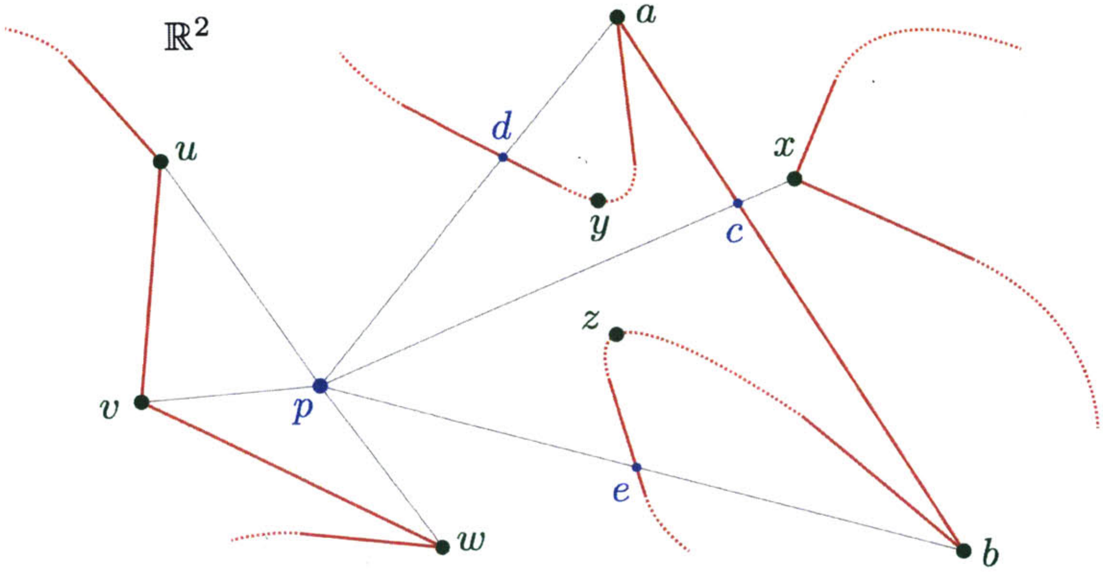


Figure 4-5: Visibility of p. If $x \in X$ is not visible from v, one of $\{a, b, y, z\} \in X$ will be.

exists a vertex $x' \in V(P) \setminus \{u, v, w\}$ not visible from and critical with v under f such that $d(\delta, x') < 0$ for all $\delta \in (0, \varepsilon]$ for some $\varepsilon > 0$. Either x' is in the infinite sector C induced by $\angle uvw$ or not. If the former, the line segment from v to x' must cross some edge (a, b) of P and $\{a, b, x', v\}$ is critical under f by Lemma 4.1.1(b). Since neither a nor b can be visible from v, then u and w must be in $\triangle abv$, and $\{u, v, w\}$ must be critical, contradicting $\{u, w\}$ contractive under f. Alternatively x' is not in C, and for every $\delta \in (0, \varepsilon]$ for some $\varepsilon > 0$, the line segment from $p(\delta)$ to x' crosses either (v, u) or (v, w). By Lemma 4.1.1(b), $d(\delta, x') \geq 0$, a contridiction, so t^* exists.

We now prove that p is visible from some $x \in X$. Suppose for contradiction that p is not visible from any $x \in X$ so that for each x there exists point $c \in \partial P$, the boundary crossing closest to p on the segment from p to x. c cannot be strictly interior to edge (v, u) or (v, w) because Lemma 4.1.1(b) implies $\|p(t^* + \delta) - x\| = \|q(t^* + \delta) - f(x)\|$ for all $d \in (0, \varepsilon]$ for some ε, a contridiction. And c cannot be v or else $\|p(t) - x\| = \|q(t) - f(x)\|$ for all $t \in [0, \ell]$. So c crosses some other edge (a, b) (see Figure 4-5). Then Lemma 4.1.1(b) implies $\|p - i\| = \|q - f(i)\|$ for $i \in \{a, b\}$, and the contrapositive of Lemma 4.1.1(a) implies for at least one vertex $i \in \{a, b\}$, $\|p(t^* + \delta) - i\| < \|q(t^* + \delta) - f(i)\|$ for all $\delta \in (0, \varepsilon]$ for some $\varepsilon > 0$. Without loss of generality assume $i = a$. Because $a \in X$, p cannot be visible from a. Let $d \in \partial P$ be the boundary crossing closest to p on the segment from p to a. There must exist some vertex y in triangle $\triangle acp$ from which p is visible because the boundary of the polygon entering the triangle at d must return to a without crossing edge (c, p). By the same argument, at least one of $\{y, b\}$ is in X, and since p is visible from y, $b \in X$. Replacing (b, e, z) for (a, d, y) in the argument above, one of $\{y, z\}$ is in X. But p is visible from both, a contradiction. □

Lemma 4.1.6. *Given polygon P with valid boundary mapping $f : \partial P \to \mathbb{R}^d$ and vertex v, contractive under f with every visible nonadjacent vertex, adjacent to two vertices $\{u, w\}$ contractive under f, a split point/image/end triple of (P, f, v) exists and can be identified in $O(d|V(P)|)$ time.*

Proof. This result follows directly by choosing any bend line/image pair of (P, f, v) according to Lemma 4.1.4, then constructing the split point/image/end triple specified by Lemma 4.1.5. Choosing a bend line/image pair can be done in $O(d)$ time. Constructing the split point/image/end triple requires a d-dimensional comparison at each vertex yielding total construction time $O(d|V(P)|)$. □

4.1.4 Partitions

To find an overall solution to the hole problem, we will repeatedly split a polygon in half, solve each piece recursively, and then join the pieces back together. Specifically, we want to find a *partition* consisting of two *partition polygons* together with respective boundary mappings such that: the partition polygons exactly cover the original polygon; the partition polygons intersect, and only on their boundaries; each partition function maps the partition polygon boundaries into the same dimensional space as the original function; the original boundary mapping of the polygon boundary is preserved by the partition functions; the intersection of the partition polygons map to the same place under both partition functions; and the partition functions are valid.

Definition 4.1.4. (Valid Partition) *Given polygon P and valid boundary mapping $f : \partial P \to \mathbb{R}^d$, define (P_1, P_2, f_1, f_2) to be a* valid partition *of (P, f) if the following properties hold:*

(1) P_1, P_2 polygons with $P = P_1 \cup P_2$; (2) $P_1 \cap P_2 = \partial P_1 \cap \partial P_2 = L \neq \emptyset$;

(3) $f_1 : \partial P_1 \to \mathbb{R}^d, f_2 : \partial P_2 \to \mathbb{R}^d$; (4) $f(p) = \begin{cases} f_1(p) & p \in \partial P \cap \partial P_1, \\ f_2(p) & \text{otherwise;} \end{cases}$

(5) $f_1(p) = f_2(p)$ for $p \in L$; (6) f_1, f_2 valid.

4.2 Algorithm

Theorem 4.2.1. *Given polygon P and boundary mapping $f : \partial P \to \mathbb{R}^d, d \geq 2$ nonstraight at finitely many boundary points, an isometric mapping $g : P \to \mathbb{R}^d$ with $g(\partial P) = f(\partial P)$ exists if and only if f is valid. A solution can be computed in polynomial time.*

The theorem implies that the necessary condition in Lemma 4.1.2 is also sufficient. Our approach is to iteratively divide P into valid partitions and combine them back together. We partition non-triangular polygons into smaller ones differently depending on which of two properties (P, f) satisfies. First we show that (P, f) satisfies at least one of these properties.

4.2.1 Existence

Lemma 4.2.1. *For every polygon P with $|V(P)| > 3$ and valid boundary mapping $f : \partial P \to \mathbb{R}^d$, either (a) there exist two nonadjacent vertices $\{u, v\}$ critical under f and visible from each other, or (b) there exists a vertex $v \in V(P)$ adjacent to two vertices $\{u, w\}$ contractive under f, or (c) both exist.*

Proof. Suppose for contradiction that there exists some (P, f) such that no two nonadjacent vertices critical under f are visible from each other and no vertex is adjacent to two vertices contractive under f. Consider any vertex v which, by the contrapositive of the latter condition, will be adjacent to two vertices $\{u, w\}$ critical under f. Since $|V(P)| > 3$, u and v are nonadjacent and cannot be visible from each other, so there must be at least one other distinct vertex x interior to $\triangle uvw$ visible from vertex v. But since x is nonexpansive with $\{u, v, w\}$ under f, $\{x, u, v, w\}$ must be critical under f, a contradiction. □

4.2.2 Constructing Partitions

Lemma 4.2.2. *Consider polygon P with valid boundary mapping $f : \partial P \to \mathbb{R}^d$ containing nonadjacent vertices $\{u, v\}$ critical under f with u visible from v. Construct polygon P_1 from the vertices of P from u to v, and P_2 from the vertices of P from v to u. Construct boundary mapping functions $f_1 : \partial P_1 \to \mathbb{R}^d$, $f_2 : \partial P_2 \to \mathbb{R}^d$ so that $f_1(x) = f(x)$ for $x \in V(P_1)$, $f_2(x) = f(x)$ for $x \in V(P_2)$, with f_1, f_2 mapping edges of P_1, P_2 to congruent line segments. Then (P_1, P_2, f_1, f_2) is a valid partition.*

Proof. Because P_1 and P_2 are constructed by splitting P along line segment $L \subset P$ from u to v, $P = P_1 \cup P_2$ and $L = P_1 \cap P_2 = \partial P_1 \cap \partial P_2$, satisfying properties (1) and (2) of a valid partition. Property (3) is satisfied by definition. Property (4) holds because f is valid, $\{u, v\}$ is critical, and points in L are nonexpansive with points in ∂P_1 and ∂P_2 by Lemma 4.1.1(a). Property (5) holds by construction. Lastly, Property (6) holds because f_1, f_2 satisfy the conditions in Lemma 4.1.3 by construction. □

Lemma 4.2.3. *Consider polygon P with valid boundary mapping $f : \partial P \to \mathbb{R}^d$ and vertex $v \in V(P)$, contractive under f with every visible nonadjacent vertex, adjacent to two vertices $\{u, w\}$ contractive under f. Let (p, q, x) be a split point/image/end triple of (P, f, v). Construct polygon P_1 from p and the vertices of P from v to x, and P_2 from p and the vertices of P from x to v. Construct boundary mapping functions $f_1 : \partial P_1 \to \mathbb{R}^d$, $f_2 : \partial P_2 \to \mathbb{R}^d$ so that $f_1(x) = f(x)$ for $x \in V(P_1) \setminus p$, $f_2(x) = f(x)$ for $x \in V(P_2) \setminus p$, $f_1(p) = f_2(p) = q$, with f_1, f_2 mapping edges of P_1, P_2 to congruent line segments. Then (P_1, P_2, f_1, f_2) is a valid partition.*

Proof. Because P_1 and P_2 are constructed by splitting P along two line segments fully contained in P, $P = P_1 \cup P_2$ and $P_1 \cap P_2 = \partial P_1 \cap \partial P_2$, satisfying properties (1) and (2) of a valid partition. Property (3) is satisfied by definition. Property (4) holds because (P, f) is valid, $V(P_1)$ and $V(P_2)$ are nonexpansive, with adjacent vertices critical under f by definition of a split point/image, and points in the new line segments are nonexpansive with points in ∂P_1 and ∂P_2 by Lemma 4.1.1(a). Property (5) holds by construction. Lastly, Property (6) holds because f_1, f_2 satisfy the conditions in Lemma 4.1.3 by construction. □

4.2.3 Triangles

Next, we establish the base case for our induction. Specifically a triangle with a valid boundary mapping of its boundary has a unique isometric mapping of its interior consistent with the provided boundary condition.

Lemma 4.2.4. *Given polygon P with $|V(P)| = 3$ and valid boundary mapping $f : \partial P \to \mathbb{R}^d$, there exists a unique isometric mapping $g : P \to \mathbb{R}^d$ such that $g(B) = f(B)$.*

Proof. Because f is valid, the vertices of P are critical under f. ∂P and $f(\partial P)$ are congruent triangles, so their convex hulls are isometric. Specifically, if P with vertices $\{u, v, w\}$ is parameterized by $P = \{p(a, b) = a(v - u) + b(w - u) + u \,|\, a, b \in [0, 1], a + b \leq 1\}$, then the affine map $g : P \to \mathbb{R}^d$ defined by

$$g(p(a, b) \in P) = a[f(v) - f(u)] + b[f(w) - f(u)] + f(u)$$

is a unique isometry for $g(B) = f(B)$. □

4.2.4 Combining Partitions

Lastly we show that we can combine isometric mappings of valid partitions into larger isometric mappings.

Lemma 4.2.5. *Consider polygon P with valid boundary mapping $f : \partial P \to \mathbb{R}^d$, with valid partition (P_1, P_2, f_1, f_2). Given isometric mappings $g_1 : P_1 \to \mathbb{R}^d$, $g_2 : P_2 \to \mathbb{R}^d$*

with $g_1(\partial P_1) = f_1(\partial P_1)$, $g_2(\partial P_2) = f_2(\partial P_2)$, *the mapping* $g : P \to \mathbb{R}^d$ *defined below is also isometric, with* $g(\partial P) = f(\partial P)$*:*

$$g(p \in P) = \begin{cases} g_1(p) & p \in P_1, \\ g_2(p) & \text{otherwise.} \end{cases}$$

Proof. First, $g(\partial P) = f(\partial P)$ because the partition is valid. Consider the shortest path K between points $p, q \in P$ composed from a finite set of line segments. Suppose for contradiction that $g(K)$ is not the same length as K. Every point in K either lies in P_1, P_2, or both by property (1) of a valid partition. Split K into a connected set of line segments, each segment fully contained in either P_1 or P_2 with endpoints in $P_1 \cap P_2$. Because g_1 and g_2 are isometric, these line segments remain the same length under g. Further, the endpoints of adjacent segments map to the same place under g_1 and g_2 by definition of a valid partition. The total length of $g(K)$ is the sum of the lengths of the intervals, the same length as K, a contradiction. □

Now we are ready to prove the theorem.

Proof. Lemma 4.1.2 implies that f is valid if g exists. We show g exists for valid f by construction. Partition (P, f) with $|V(P)| > 3$ as follows. If (P, f) contains two nonadjacent vertices $\{u, v\}$ critical under f and visible from each other, divide using Routine 1: partition using the construction in Lemma 4.2.2. Otherwise divide using Routine 2: partition using the construction in Lemma 4.2.3, applying Routine 1 to each partitioned polygon immediately after. Note that both polygons generated by the construction from Lemma 4.2.3 are guaranteed to contain two nonadjacent vertices critical under f and visible from each other, namely $\{u, p\}$ and $\{w, p\}$, so each can be divided using Routine 1. Recursively fill each partitioned polygon with an isometric mapping of their interior and combine them into a mapping $g : P \to \mathbb{R}^d$ using the construction in Lemma 4.2.5. Since the partitions are valid, g is isometric with $g(\partial P) = f(\partial P)$. Construct isometries for triangular polygons, the base case of the recursion, according to Lemma 4.2.4.

To show the recursion terminates, consider state i where P is partitioned into a set of n_i polygons $\mathcal{P}_i = \{P_1, \ldots, P_{n_i}\}$. Define potential $\Phi_i = \sum_{P_j \in \mathcal{P}_i}(|V(P_j)| - 3)$ with $\Phi_0 = |V(P)| - 3$. Partitioning a polygon using Routine 1 yields state $i+1$ with $\Phi_{i+1} = \Phi_i - 1$: Lemma 4.2.2 adds two vertices, the number of polygons increases by one, and $2-3=-1$. Partitioning a polygon using Routine 2 also yields $\Phi_{i+1} = \Phi_i - 1$: Lemma 4.2.3 adds four vertices, Lemma 4.2.2 adds two vertices with each application, the number of polygons increases by three, and $4+2\times 2-3\times 3=-1$. Lemma 4.2.1 ensures that one of the routines can always be applied to any non-triangular polygon. When $\Phi_i = 0$, all partitioned polygons are triangles and no polygon can be partitioned further. The iteration terminates after Φ_0 calls to either routine.

Let n be the number of vertices $|V(P)|$ in the input polygon. At the start of the algorithm, all critical vertex pairs can be identified naively in $O(dn^2)$ time. Application of either routine requires at most $O(dn)$ time, and both routines can update and maintain new critical vertex pairs in partition polygons at no additional cost. Each routine is called no more than $O(n)$ times. Only a linear number of triangles are produced and the construction of each g_i takes constant time. The running time of the entire construction is thus $O(dn^2)$, which is polynomial. □

4.2.5 Edge Insetting

The algorithm described above terminates in quadratic time because at each step, the hole is split either along a critical path or by insetting a single crease from a boundary vertex. Of course not all isometries satisfying a boundary condition can be generated using such a procedure. Figure 4-6 shows a few examples of some isometries for which no folded boundary point has only a single crease adjacent to it, so the algorithm described would not be able to find these solutions.

Insetting a vertex v to a split point p has the property that p is critical with vertex v, both vertices u and w adjacent to v, and some fourth vertex x visible from p; see Figure 4-7. Thus, the triangles $\triangle vpw$ and $\triangle f(v)f(p)f(w)$ are congruent. Instead of insetting v all the way to point p, we can instead choose to inset v to some point q on the interior of $\triangle vpw$, with $f(q)$ mapping to the analogous point in $\triangle f(v)f(p)f(w)$.

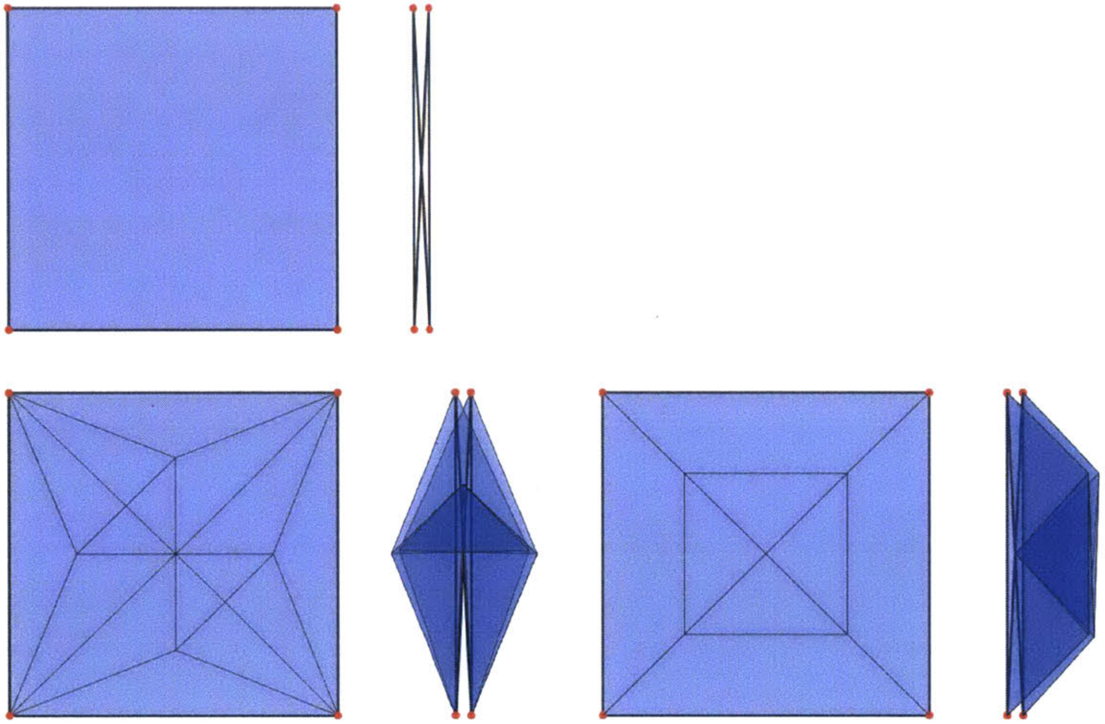


Figure 4-6: Not all isometries are accessible with vertex insetting. Here are two simple crease patterns that cannot be generating using vertex insetting. The first is inaccessible because no folded point on the boundary has only a single crease adjacent to it. While the second example has a single crease adjacent to each folded point on the boundary, none are inset all the way to a split point with another boundary point.

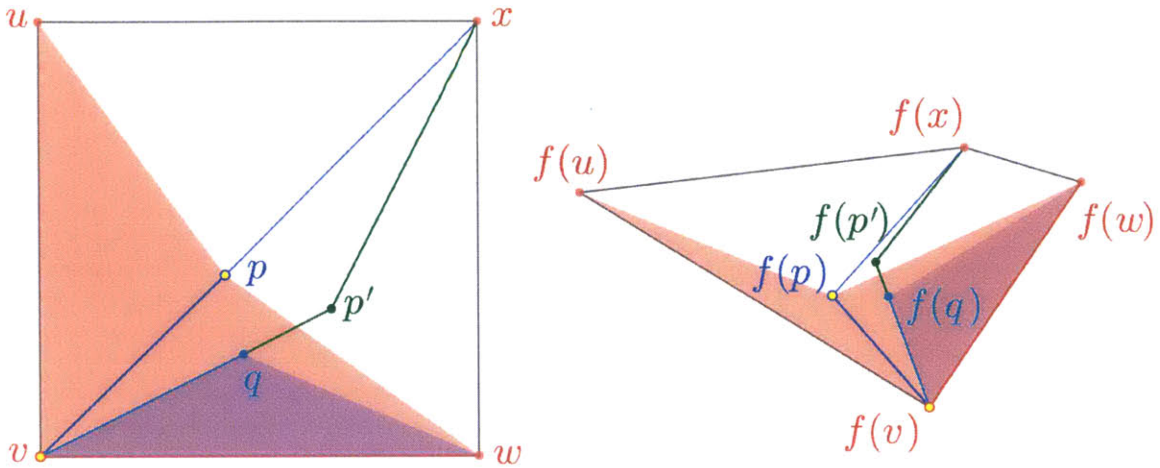


Figure 4-7: Insetting from an edge to a non-split point.

In fact, point q and $f(q)$ simply define a new inset direction for vertex v. Insetting v along line vq will eventually become critical with some third vertex x' resulting in a new split point p', critical with v, w, and x' but not critical with u.

Of course we need not inset all the way to p'. We may instead choose to inset in the direction of vq and $f(v)f(q)$ by any distance s up to distance $|vp'|$. This choice of p, q, and s defines a three-dimensional space of solutions. Choosing a point in this solution space together with the points v and w define the set of all possible triangles that may be adjacent to edge vw in any isometry. Choosing such a point and partitioning the boundary into the triangle $\triangle vpw$ and the remainder, we call this process *edge insetting.* Edge insetting breaks up the original polygon into two nonexpansive boundaries that form a valid partition. We can then repeat the procedure until the entire hole is filled. Given a target isometry satisfying the boundary condition, we can simply paint the triangles from the boundary inward using edge insetting. This level of flexibility yields a very permissive procedure to explore satisfying isometries, and the following universality result.

Theorem 4.2.2. *Given a paper and a nonexpansive folding of its boundary, every isometry of the paper satisfying the boundary condition can be constructed by edge insetting.*

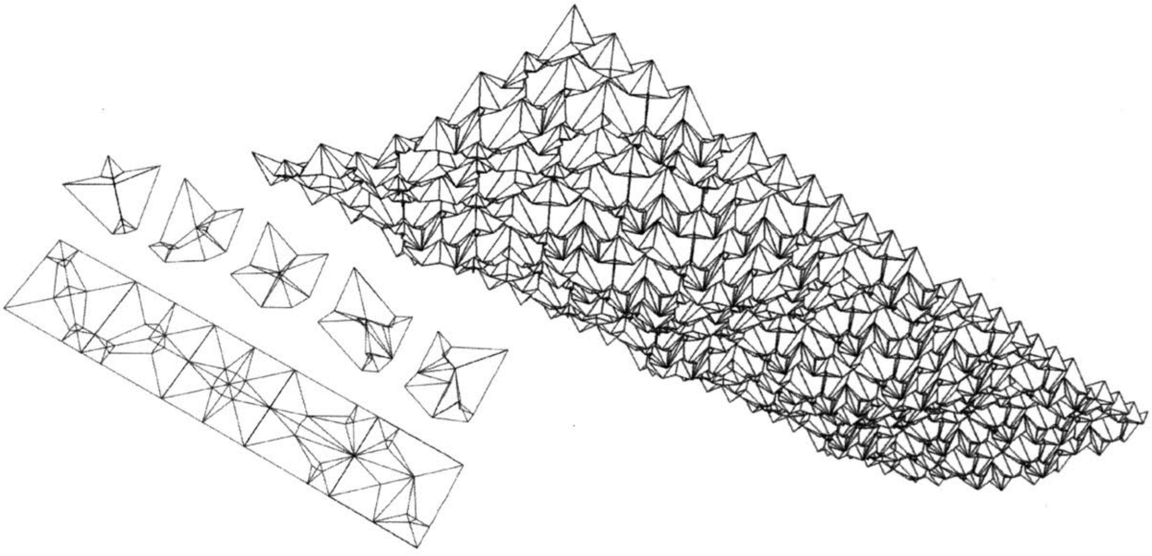


Figure 4-8: A three-dimensional abstract tessellation formed by tiling five different square units, each corner in either a binary low or high state. Units were designed using this algorithm having common boundaries, connected to form single sheet tessellations.

Of course in general, a sequence of edge insetting procedures does not have to terminate. Each time we use the edge inset procedure, we will not have reduced the complexity of the boundary; indeed, the number of boundary vertices typically increases by one. However not all isometries satisfying a folding of the boundary have finite complexity, so a procedure that has the power to construct such isometries cannot hope to complete in finite time.

4.3 Implementations

Much of the intuition for this algorithm was developed while working on the design of various three-dimensional tessellations, specifically while working on Maze Folding [15] and a private commission designing an origami chandelier for Moksa, a restaurant in Cambridge, MA (see Figure 4-8). A version of this algorithm was implemented for flat folds ($d = 2$) in 2010 using MATLAB (see Figure 4-9).

In 2015 we wrote a program to implement the full algorithm for isometries in three dimensions ($d = 3$). The program was written in coffeescript and can be found at

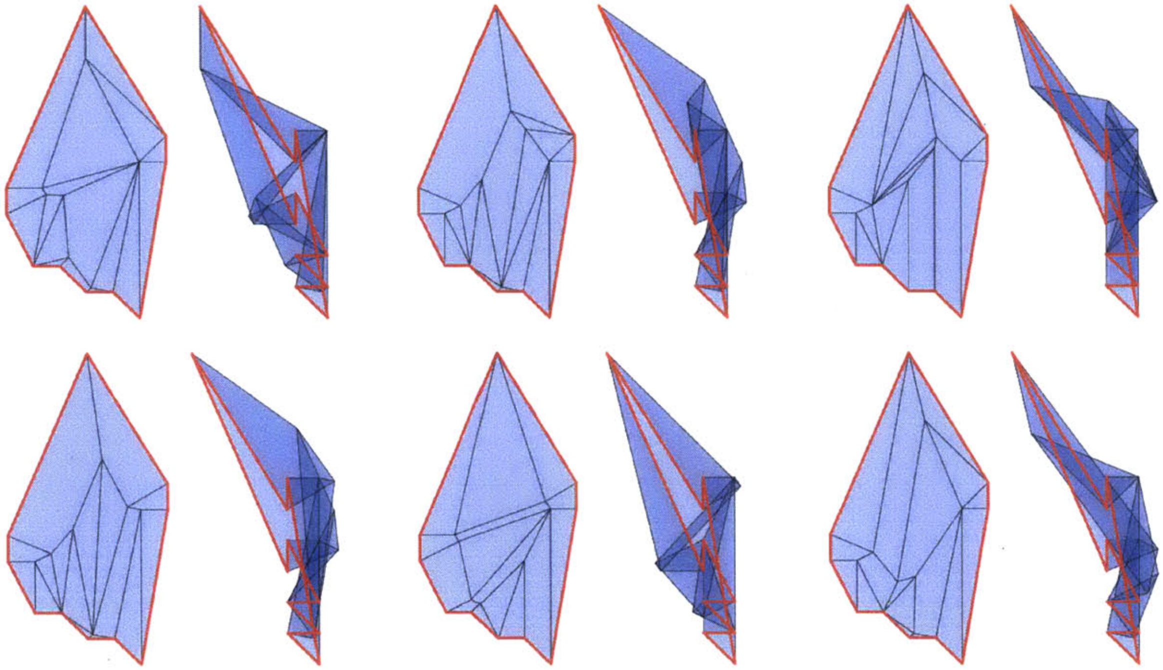


Figure 4-9: Various solutions for the same input polygon and boundary mapping found by our MATLAB implementation for $d = 2$.

http://jasonku.scripts.mit.edu/hole. The input can be specified as an OBJ file and the user can step through the solution space, making decisions at each step of the algorithm. The polynomial time algorithm can be accessed by pressing spacebar to find a single solution, but both vertex insetting and edge insetting can be accessed via the interactive controls. Figure 4-10 shows a screenshot of the implementation.

4.4 Remarks

We have proposed an algorithm for finding isometric mappings consistent with prescribed boundary mappings that runs in polynomial time. This algorithm was inspired by the universal molecule construction; instead of insetting an input polygon perimeter at a constant rate from all edges at once, our algorithm insets each vertex serially as far as possible. Our construction cannot find all possible isometric solutions, though the algorithm provides a rich family of solutions given choice of bend line and image with each application of Routine 2: two choices when $d = 2$ and an

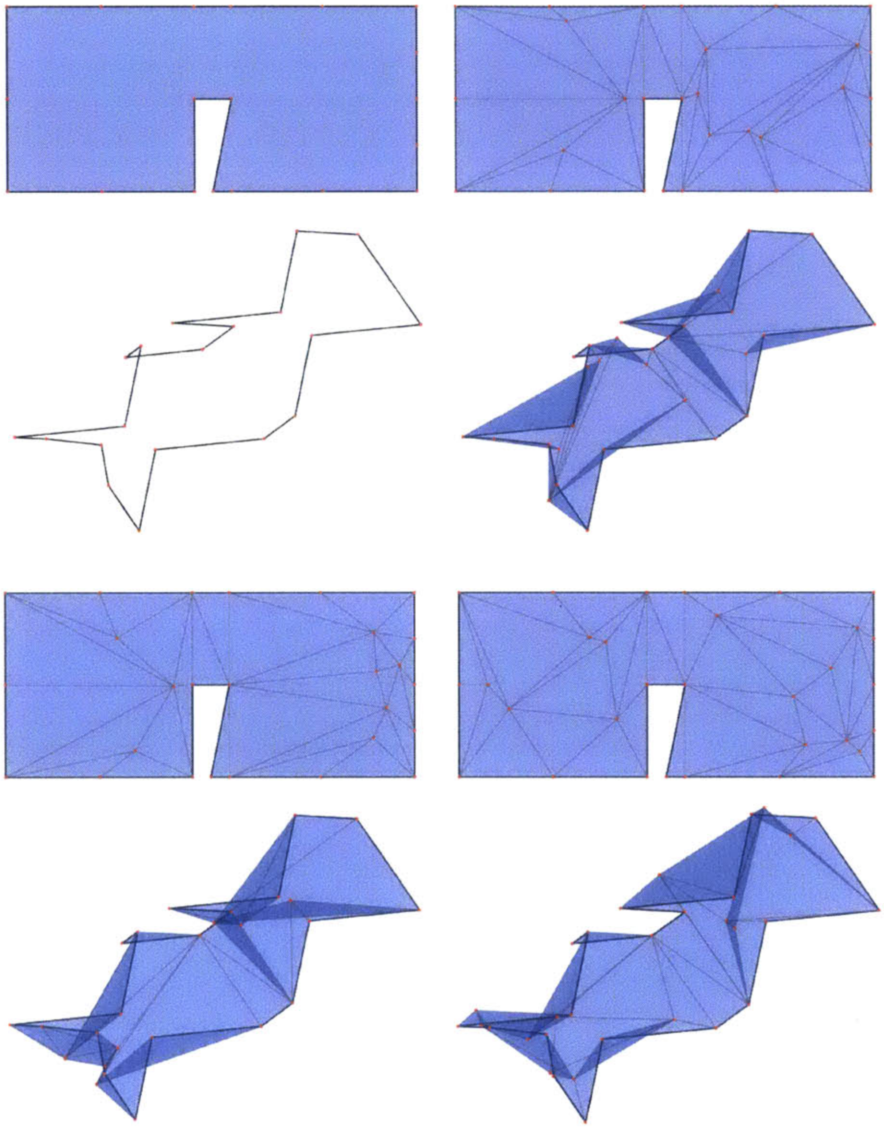


Figure 4-10: A screenshot of the hole filling implementation in action. The top left image shows the input flat paper with a nonexpansive boundary folding in three dimensions. The remaining figures show three solutions found by the algorithm using a random search through the configuration space, showing the crease pattern as well as the isometry satisfying the boundary condition.

infinite set of choices for $d > 2$. This algorithm can be generalized by not insetting vertices all the way to split points, and by solving vertices locally with more than one crease at a vertex. We conjecture that adding such flexibility would allow construction of the entire space of isometric solutions following a similar procedure to our construction.

Recall that the proposed algorithm does not address self intersection and cannot guarantee the existence of a valid layer ordering for the isometries found; however, because the space of solutions is large for a generic input, one might be able to construct non-self-intersecting solutions by directing the algorithm's decisions appropriately through the solution space. Additionally, the proposed algorithm only addresses instances for f folded at finitely many points. It is conceivable that a similar algorithm could be used to design curved foldings. We leave these as open problems.

4.5 Folded Quadrilateral Boundaries

In this section, we analyze single crease patterns that can fold to multiple prescribed folded boundaries and flat-foldable states. We restrict our study to folding two dimensional paper into two and three dimensions.

4.5.1 Flat Foldability

Theorem 4.5.1. *Given a four cornered paper, the exists a single vertex crease pattern folding through each corner of the paper that folds flat.*

Proof. A single degree-four vertex in a flat-foldable crease pattern must obey Kawasaki's theorem, that the sum of opposite angles sum to π. From this condition, one can derive a condition on possible positions (x, y) of the single vertex. We can parameterize any simple quadrilateral with cyclically ordered points $a = (-1, 0)$, $b = (x_1, y_1)$, $c = (1, 0)$, and $d = (x_2, y_2)$, where y_1 is positive and y_2 is negative, and the line from a to c is a visible diagonal. The condition on the location of a flat-foldable vertex is then given by the following cubic equation:

$$\begin{aligned} &x(y_1 + y_2)(x^2 + y^2 - 1) - y(x_1 + x_2)(x^2 + y^2 + 1) + \\ &\quad (x_1 y_2 + x_2 y_1)(y^2 - x^2 + 1) + 2xy(1 + x_1 x_2 - y_1 y_2) = 0. \end{aligned} \tag{4.1}$$

The curve defined by this equation passes through each corner of the paper, as can be readily verified. However, we must prove that the curve passes through the interior of the paper. It suffices to show that the tangent to the curve at one of the vertices passes between its two adjacent edges. Taking partial derivatives of Equation 4.1, one can show the tangent to the curve at p_a has the same direction the following vector:

$$v_T = (\,(x_1 + 1)(x_2 + 1) - y_1 y_2,\; y_1(x_2 + 1) + y_2(x_1 + 1)\,)\,. \tag{4.2}$$

The edges adjacent to p_a have directions $v_b = (x_1 + 1, y_2)$ and $v_d = (x_2 + 1, y_2)$ respectively. Taking magnitude of the cross products in the $\hat{z}$ direction out of the plane yields the following relations:

$$(v_T \times v_b) \cdot \hat{z} = -(\,(1 + x_1)^2 + y_1^2\,)y_2; \tag{4.3}$$

$$(v_T \times v_d) \cdot \hat{z} = -(\,(1 + x_2)^2 + y_2^2\,)y_1. \tag{4.4}$$

Because y_2 is always negative, the first condition is always positive, so the top edge is a left turn from the tangent line. Because y_1 is always positive, the second condition is always negative, so the bottom edge is a right turn from the tangent line, so local to p_a, the curve must intersect the quadrilateral, completing the proof. □

In the special case of a kite quadrilateral where each edge is adjacent to an edge of the same length, Equation 4.1 degenerates into a line and circle. This can be shown by imposing the condition $x_1 = x_2$ and $y_1 = -y_2$. The result is as follows:

$$y\left(\left(x - \frac{1 + x_1^2 + y_1^2}{2x_1}\right)^2 + y^2 + 1 - \left(\frac{1 + x_1^2 + y_1^2}{2x_1}\right)^2\right) = 0. \tag{4.5}$$

Even more special for a square when $x_1 = x_2 = 0$ and $y_1 = -y_2 = 1$, the solution space is simply the horizontal and vertical lines:

$$xy = 0. \tag{4.6}$$

4.5.2 Boundary Condition and Flat Foldability

Now we turn to satisfying boundary conditions and flat foldability. The previous section proved that there always exists isometries that satisfy a set of boundary conditions. For quadrilaterals, the solution space of single vertex crease patterns satisfying a folding of the boundary results in an ellipse on the interior of the paper. The equation of this ellipse in general is quite complicated. However, in the case of kite quadrilaterals, the ellipse is axis aligned with the diagonals, and for squares the ellipse is center is fixed, and has a particularly simple form. Let the corners of the square be $(-1, 0)$, $(0, 1)$, $(1, 0)$ and $(0, -1)$. We parameterize the folding of the square boundary by the distances between the two diagonals, distance $2\sqrt{1-a^2}$ along the x axis and distance $2\sqrt{1-b^2}$ along the y axis; this parameterization will simplify the equations later on. Using this parameterization, the equation of the ellipse of crease pattern vertices satisfying the boundary condition (a, b) is as follows:

$$\frac{x^2}{a^2(1-b^2)} + \frac{y^2}{b^2(1-a^2)} = a^2 + b^2. \tag{4.7}$$

Figure 4-11 shows ellipses corresponding to two different boundary conditions. Since the ellipse is centered at the origin, it must cross the x and y axes four times except in the degenerate cases where the ellipse becomes a line or a point, specifically when a or b equal 1 or 0. When both a and b equal 0, the square is flat in the folding. When a or b equals 0, the diagonal is critical and the isometry is unique. When a or b equals 1, a line segment of solutions exist. And when both a and b equal 1, the only solution is the center of the paper. Thus we arrive at the following:

Theorem 4.5.2. *Given a square of paper and a folding of its boundary folded only at the corners so that opposite corners are contractive under the folding and not the same point, there are exactly four single-vertex flat-foldable crease patterns satisfying the boundary condition.*

If we choose a crease pattern by fixing (x, y), Equation 4.7 defines a quartic function on a and b defining the one degree of freedom mechanism of the crease pattern.

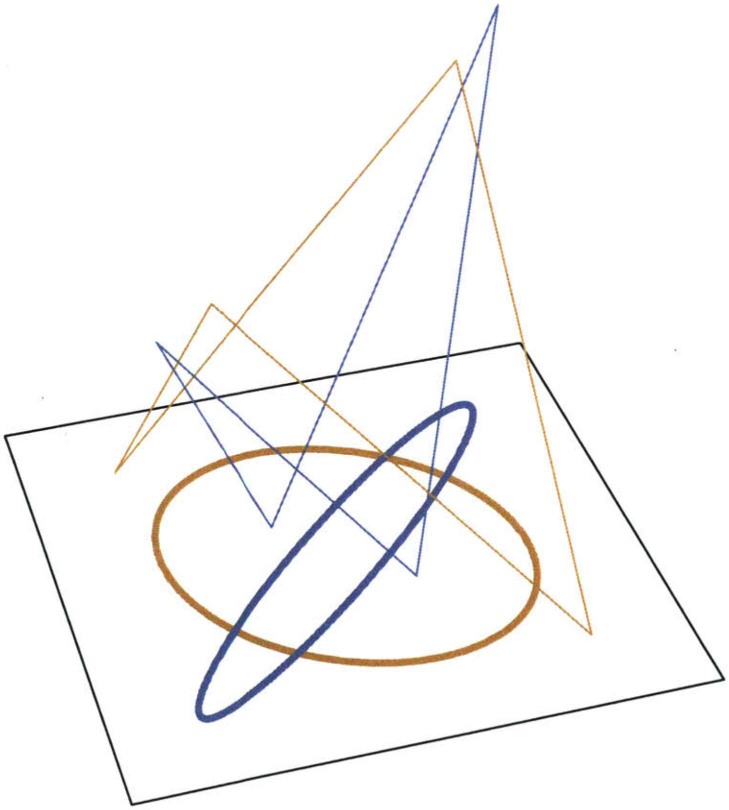


Figure 4-11: Ellipses corresponding to the space of possible single vertex locations corresponding to crease patterns that fold to the corresponding boundaries.

Figure 4-12 shows a plot of this state space for one such crease pattern. The bottom right corner corresponds to the flat state. The bottom curve

4.5.3 Two Boundary Conditions

Now we wish to examine crease patterns that can fold to two different boundary conditions at the same time. For simplicity, we examine squares, using the same parameterization as before.

Single Vertex Crease Patterns

Theorem 4.5.3. *Given a square of paper and two foldings (a_1, b_1), (a_2, b_2) of its boundary folded only at the corners, then if the intervals $[a_1, b_1]$ and $[a_2, b_2]$ overlap, then there exists a single vertex crease pattern that folds exactly to both boundaries.*

Proof. The proof is by construction. The approach will be to calculate the set of possible crease patterns with one interior vertex that folds to each boundary, and show that the two sets have crease patterns in common when the intervals $[a_1, b_1]$ and $[a_2, b_2]$ overlap.

We have already shown that the solution space for each boundary condition is an ellipse given by Equation 4.7. Given two such ellipses parameterized by (a_1, b_1) and (a_2, b_2), they can be made to intersect as long as the smaller major radius is larger than the minor radius of the other since the roles of a and b are interchangeable under boundary mappings. Specifically, the following must hold:

$$\begin{aligned}&\left(\left(1-a_1^2\right)\left(\left(1-b_2^2\right)-\left(1-b_1^2\right)\left(a_2^2+b_2^2\right)\right)-\left(1-a_2^2\right)\left(\left(1-b_1^2\right)-\left(1-b_2^2\right)\left(a_1^2+b_1^2\right)\right)\right)\\&\left(\left(1-b_1^2\right)\left(\left(1-a_2^2\right)-\left(1-a_1^2\right)\left(a_2^2+b_2^2\right)\right)-\left(1-b_2^2\right)\left(\left(1-a_1^2\right)-\left(1-a_2^2\right)\left(a_1^2+b_1^2\right)\right)\right)\leq 0.\end{aligned} \tag{4.8}$$

A simple yet tedious case analysis shows that this equation holds when intervals $[a_1, b_1]$ and $[a_2, b_2]$ overlap. The converse statement is not true as there are points when the intervals do not overlap such that the inequality is still true. Figure 4-13 depicts the relevant regions of the configuration space over a_1, b_1, a_2, b_2. □

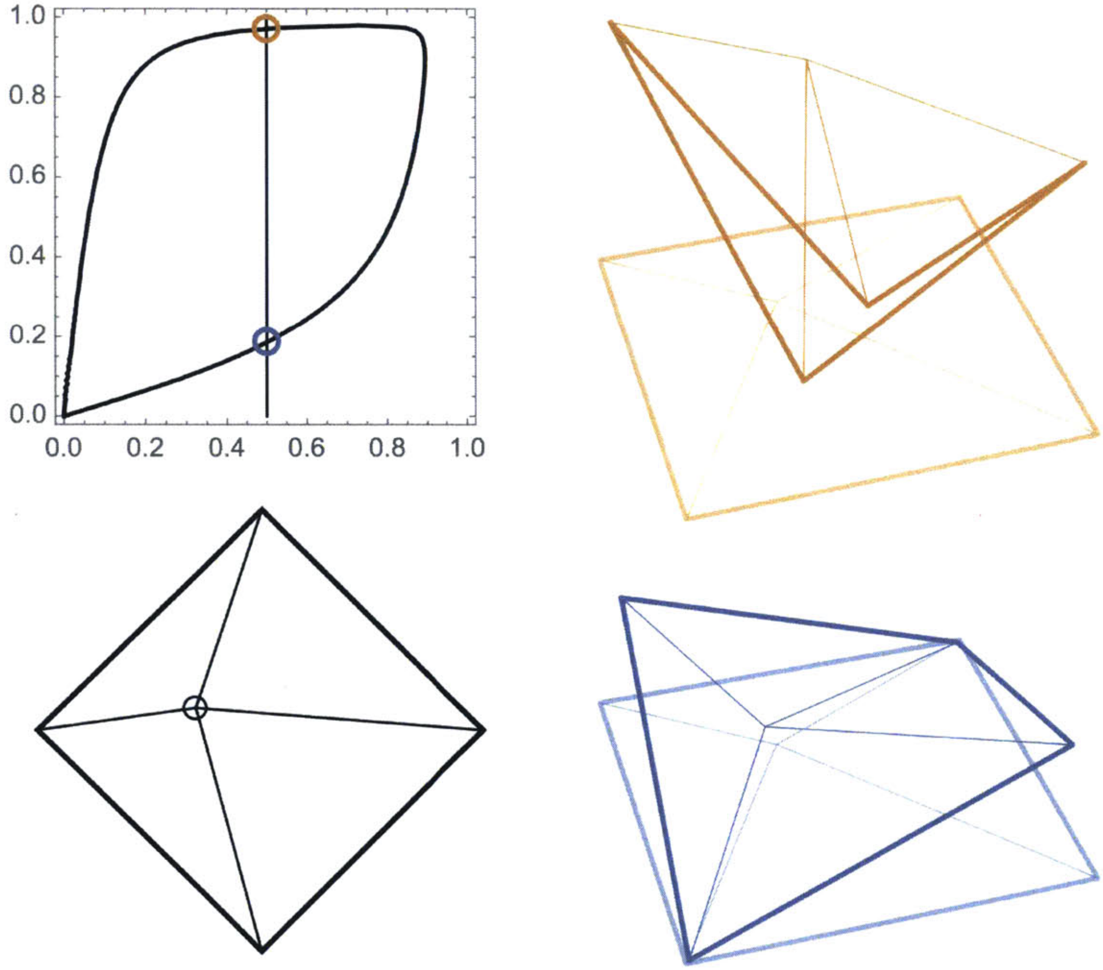


Figure 4-12: The state space of a single vertex crease pattern, plotting a vs. b, and two folded states for a single value of a. The orange state has a high value of b so the top and bottom points are close together. The blue state has a low value of b so the bottom points are far apart.

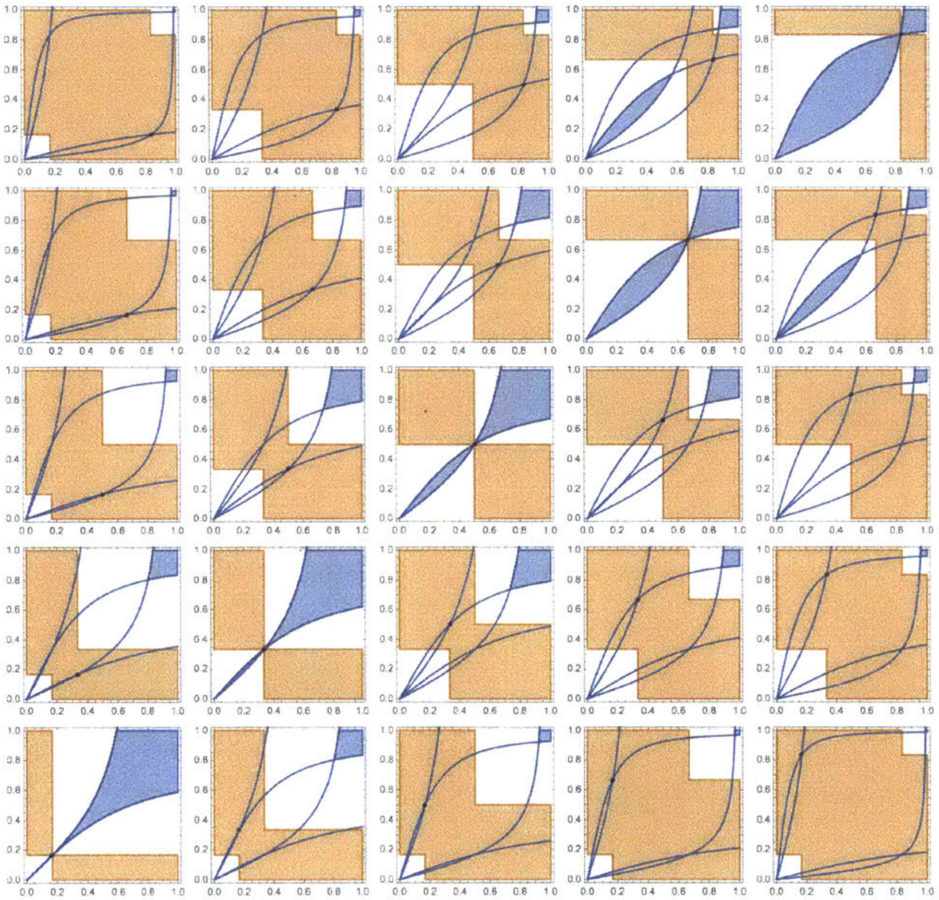


Figure 4-13: Plots showing regions where single vertex crease patterns do not exist. For a fixed (a_1, b_1), the plot at the corresponding coordinate ranges over possible values of (a_2, b_2). The orange region represents the values of (a_2, b_2) such that the intervals $[a_1, b_1]$ and $[a_2, b_2]$ overlap, as provided by the claim. The blue region indicates the values of (a_2, b_2) that prescribe an ellipse which completely surrounds the ellipse prescribed by (a_1, b_1) or lies completely inside the ellipse prescribed by (a_1, b_1). The white region indicates values not covered by the rule of thumb that actually do admit a solution.

Two-Vertex Crease Patterns

Theorem 4.5.4. *Given any two nonexpansive boundary foldings of a square paper folding at its vertices, there exists a one or two-vertex crease pattern that can fold rigidly to meet both boundary conditions.*

Proof. The proof is by construction. The approach will be to calculate a subset of possible crease patterns with two interior vertices that folds to each boundary, and show that the two sets have crease patterns in common.

We will parameterize a subset of two-vertex crease patterns in the special case where one vertex resides on a diagonal. We will let s be the distance between this vertex p and point $(-1, 0)$. Solving the distance equations again yields the equation of an ellipse, this time of the following form:

$$\frac{(x - x_0)^2}{r_x^2} + \frac{y^2}{r_y^2} - 1 = 0. \tag{4.9}$$

However, this time there are two possible ellipses for each choice of boundary condition: one when the crease from $(-1, 0)$ to p is a valley fold, and one when the crease is a mountain fold. The parameters of the ellipse in each case are given by:

$$x_0 = \frac{sb^2(1 + b^2)}{2(a^2 + b^2)((1 - s) + b^2) \pm 2sb\sqrt{1 - a^2}\sqrt{a^2 + b^2}}; \tag{4.10}$$

$$r_x = b\sqrt{\frac{1 - a^2}{a^2 + b^2}} - \frac{sb^2(1 - b^2)}{2(a^2 + b^2)((1 - s) + b^2) \pm 2sb\sqrt{1 - a^2}\sqrt{a^2 + b^2}}; \tag{4.11}$$

$$r_y = a\sqrt{\frac{1 - b^2}{a^2 + b^2}}\left(1 - \frac{sb^2}{(a^2(1 - s) + b^2) \pm b\sqrt{1 - a^2}\sqrt{a^2 + b^2}}\right). \tag{4.12}$$

When $s = 0$, these parameters reduce Equation 4.9 to Equation 4.7. Figure 4-14 shows a plot of this state space for one such crease pattern. The bottom right corner corresponds to the flat state. The bottom curve

The equations continue to define an ellipse as long as r_y does not become negative. If the ellipse corresponding to (a_1, b_1) and (a_2, b_2) do not intersect at $s = 0$, then that means both r_x and r_y are larger for one and not the other because x_0 is zero. Without loss of generality, assume $a_1 > a_2$ and $b_1 > b_2$. r_y is zero precisely when:

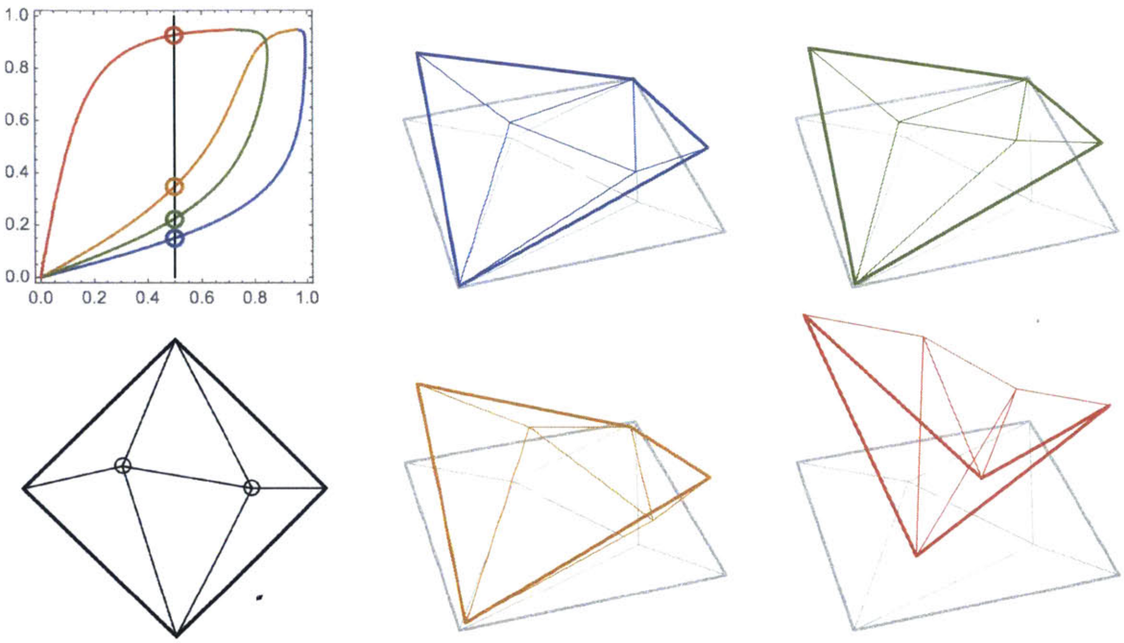


Figure 4-14: The state space of a two vertex crease pattern, plotting a vs. b, and four folded states for a single value of a.

$$s(r_y = 0) = 1 \pm b\sqrt{\frac{1-a^2}{a^2+b^2}}. \tag{4.13}$$

Since r_y is symmetric about 1 for any (a, b), this means r_y for (a_1, b_1) and r_y for (a_2, b_2) must be equal form some s. If they are equal, their corresponding ellipses must intersect, which corresponds to a two vertex solution. Figure 4-15 shows how x_0, r_x, and r_y vary with respect to s. Figure 4-16 shows a crease pattern that folds to two prescribed boundary conditions. □

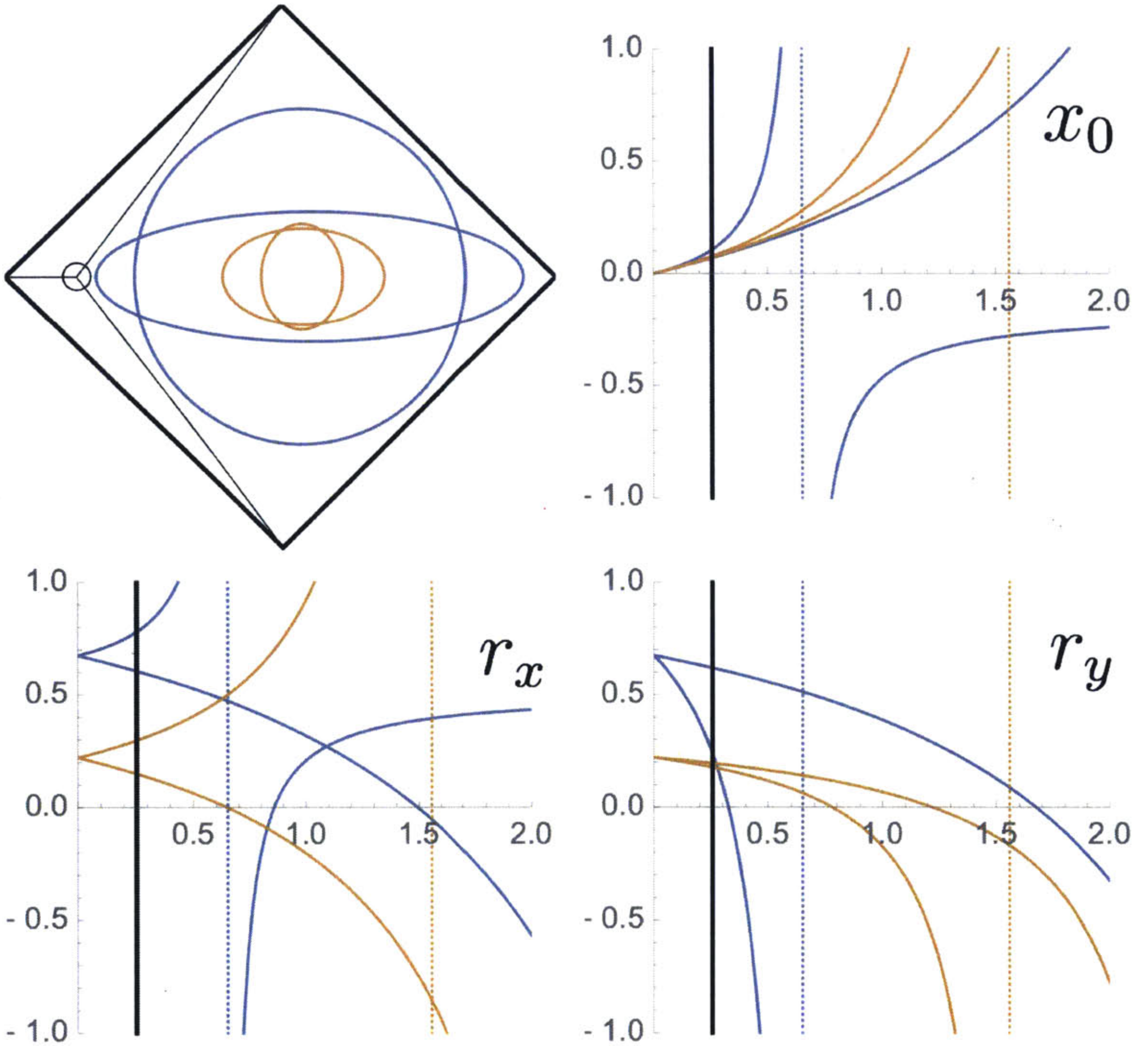


Figure 4-15: Graphs of how x_0, r_x, and r_y vary with respect to s for $(a_1, b_1) = (0.3, 0.3)$ (blue) and $(a_2, b_2) = (0.95, 0.95)$ (orange).

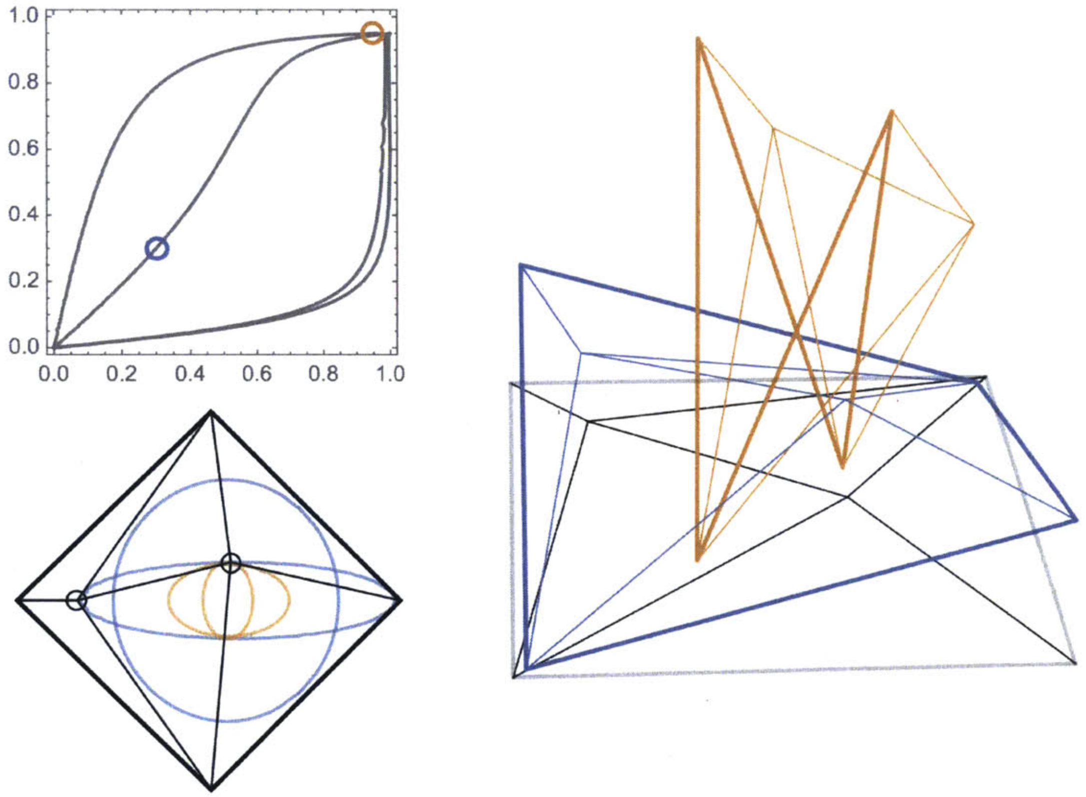


Figure 4-16: A crease pattern that satisfies two boundary conditions for $(a_1, b_1) = (0.3, 0.3)$ (blue) and $(a_2, b_2) = (0.95, 0.95)$ (orange), along with the foldings that satisfy the constraints.

Chapter 5

Conclusion

This thesis has sought to further the dream of a transformable world. Folding is an important mechanism that must be harnessed to make the modular, reusable, transformable structures and devices of the future. I have presented fundamental analysis into key problems in computing foldability, both for box-pleated crease patterns and simple folding motions in many models, motivating the design of surfaces with more structure and coherence. I have shown how one can compensate for material volume when building transformable surfaces from real materials. And I have presented a very general method for designing folded surfaces from one or multiple boundary constraints. While there is still much work to be done in order to achieve the vision of real transformers, I hope that the work in this thesis can be expanded upon so that we can make many more beautiful and efficient, complex and transformable structures far into the future.

Bibliography

[1] Hugo A. Akitaya, Kenneth C. Cheung, Erik D. Demaine, Takashi Horiyama, Thomas C. Hull, Jason S. Ku, Tomohiro Tachi, and Ryuhei Uehara. Box pleating is hard. In *The 18th Japan Conference on Discrete and Computational Geometry and Graphs*, 2015.

[2] Hugo A Akitaya, Erik D Demaine, and Jason S Ku. Simple folding is strongly np-complete. In *Fall Workshop on Computational Geometry*, 2014.

[3] Ichiro Ario, Masatoshi Nakazawa, Yoshikazu Tanaka, Izumi Tanikura, and Syuichi Ono. Development of a prototype deployable bridge based on origami skill. *Automation in Construction*, 32:104–111, 2013.

[4] 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. Preliminary results published in the Proceedings of WADS 2001.

[5] William J Arora, Hyun Jin In, Tilman Buchner, S Yang, Henry I Smith, and George Barbastathis. Nanostructured origami$^{\text{TM}}$ 3d fabrication and self assembly process for soldier combat systems. *Selected Topics in Electronics and Systems*, 42:473, 2006.

[6] D. Balkcom. *Robotic Origami Folding.* PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, August 2002.

[7] Nadia M. Benbernou, Erik D. Demaine, Martin L. Demaine, and Aviv Ovadya. Universal hinge patterns to fold orthogonal shapes. In *Origami^5: Proceedings of the 5th International Conference on Origami in Science, Mathematics and Education (OSME 2010)*, pages 405–420. A K Peters, Singapore, July 13–17 2010.

[8] 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, 2002.

[9] Marshall Bern and Barry Hayes. The complexity of flat origami. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '96,

pages 175–183, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.

[10] John C Bowers and Ileana Streinu. Lang's universal molecule algorithm. *Annals of Mathematics and Artificial Intelligence*, pages 1–30, 2015.

[11] Yan Chen, Rui Peng, and Zhong You. Origami of thick panels. *Science*, 349(6246):396–400, 2015.

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

[13] Christoffer Cromwik. *Numerical folding of airbags based on optimization and origami.* PhD thesis, Chalmers University of Technology, 2007.

[14] N. De Temmerman, L. Alegria Mira, A. Vergauwen, H. Hendrickx, and W. P. De Wilde. Transformable structures in architectural engineering. *High Performance Structures and Materials VI*, 124:457, 2012.

[15] Erik D. Demaine, Martin L. Demaine, and Jason Ku. Folding any orthogonal maze. In *Origami^5: Proceedings of the 5th International Conference on Origami in Science, Mathematics and Education (OSME 2010)*, pages 449–454. A K Peters, Singapore, July 13–17 2010.

[16] Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Folding and cutting paper. In *Revised Papers from the Japan Conference on Discrete and Computational Geometry (JCDCG'98)*, volume 1763 of *Lecture Notes in Computer Science*, pages 104–117, Tokyo, Japan, December 9–12 1998.

[17] Erik D. Demaine, David Eppstein, Adam Hesterberg, Hiro Ito, Anna Lubiw, Ryuhei Uehara, and Yushi Uno. Folding a paper strip to minimize thickness. In *Proceedings of the 9th International Workshop on Algorithms and Computation (WALCOM 2015)*, volume 8973 of *Lecture Notes in Computer Science*, pages 113–124, Dhaka, Bangladesh, February 26–28 2015.

[18] Erik D. Demaine, Sándor P. Fekete, and Robert J. Lang. Circle packing for origami design is hard. In *Origami^5: Proceedings of the 5th International Conference on Origami in Science, Mathematics and Education (OSME 2010)*, pages 609–626. A K Peters, Singapore, July 2010.

[19] Erik D. Demaine and Jason S. Ku. Filling a hole in a crease pattern: Isometric mapping from prescribed boundary folding. In *Origami^6: I. Mathematics*, pages 177–188. American Mathematical Soc., 2015.

[20] Erik D. Demaine and Joseph O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra.* Cambridge University Press, July 2007.

[21] Shawn M Douglas, Hendrik Dietz, Tim Liedl, Björn Högberg, Franziska Graf, and William M Shih. Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature*, 459(7245):414–418, 2009.

[22] James T. Early, Roderick Hyde, and Richard L. Baron. Twenty-meter space telescope based on diffractive Fresnel lens. In *Proc. The International Society for Optics and Photonics 5166, UV/Optical/IR Space Telescopes: Innovative Technologies and Concepts*, pages 148–156, 2004.

[23] Bryce J. Edmondson, Robert J. Lang, Spencer P. Magleby, and Larry L. Howell. An offset panel technique for rigidly foldable origami. In *Proceedings of ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.*, page V05BT08A054, 2014.

[24] E. A. Elsayed and Basily B. Basily. A continuous folding process for sheet materials. *International Journal of Materials and Product Technology*, 21(1–3):217–238, 2004.

[25] S. M. Felton, M. T. Tolley, C. D. Onal, D. Rus, and R. J. Wood. Robot self-assembly by folding: A printed inchworm robot. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 277–282, May 2013.

[26] T. Fleischmann, K. Kubota, P. O. Vaccaro, T.-S. Wang, S. Saravanan, and N. Saito. Self-assembling gaas mirror with electrostatic actuation using micro-origami. *Physica E: Low-dimensional Systems and Nanostructures*, 24(1–2):78 – 81, 2004. Proceedings of the International Symposium on Functional Semiconductor Nanostructures 2003.

[27] H. C. Greenberg, M. L. Gong, S. P. Magleby, and L. L. Howell. Identifying links between origami and compliant mechanisms. *Mechanical Sciences*, 2(2):217–225, 2011.

[28] E. Hawkes, B. An, N. M. Benbernou, H. Tanaka, S. Kim, E. D. Demaine, D. Rus, and R. J. Wood. Programmable matter by folding. *Proceedings of the National Academy of Sciences*, 107(28):12441–12445, 2010.

[29] C. Hoberman. Reversibly expandable structures, January 1 1991. US Patent 4,981,732.

[30] C. Hoberman. Folding structures made of thick hinged sheets, September 14 2010. US Patent 7,794,019.

[31] Alan Huang. Computational origami: the folding of circuits and systems. *Appl. Opt.*, 31(26):5419–5422, Sep 1992.

[32] D.A. Huffman. Curvature and creases: A primer on paper. *IEEE Transactions on Computers*, 25(10):1010–1019, 1976.

[33] Thomas Hull. The combinatorics of flat folds: a survey. In *Third International Meeting of Origami Science*, pages 29–38, 2002.

[34] Mustapha Jamal, Sachin S. Kadam, Rui Xiao, Faraz Jivan, Tzia-Ming Onn, Rohan Fernandes, Thao D. Nguyen, and David H. Gracias. Tissue engineering: Bio-origami hydrogel scaffolds composed of photocrosslinked peg bilayers. *Advanced Healthcare Materials*, 2(8):1066–1066, 2013.

[35] S. M. Jurga, C. H. Hidrovo, J. Niemczura, H. I. Smith, and G. Barbastathis. Nanostructured origami. In *Third IEEE Conference on Nanotechnology*, volume 1, pages 220–223 vol.2, Aug 2003.

[36] Jason S. Ku and Erik D. Demaine. Folding flat crease patterns with thick materials. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V05BT08A056–V05BT08A056. American Society of Mechanical Engineers, 2015.

[37] Kaori Kuribayashi, Koichi Tsuchiya, Zhong You, Dacian Tomus, Minoru Umemoto, Takahiro Ito, and Masahiro Sasaki. Self-deployable origami stent grafts as a biomedical application of ni-rich tini shape memory alloy foil. *Materials Science and Engineering: A*, 419(1–2):131–137, 2006.

[38] Robert J. Lang. A computational algorithm for origami design. In *Proc. 12th Symp. Computational Geometry*, pages 98–105, Philadelphia, PA, May 1996.

[39] Robert J. Lang and Erik D. Demaine. Facet ordering and crease assignment in uniaxial bases. In *Origami^4: Proceedings of the 4th International Meeting of Origami Science, Math, and Education (OSME 2006)*, pages 189–205. A K Peters, Pasadena, California, September 8–10 2006.

[40] W Liu and K Tai. Optimal design of flat patterns for 3d folded structures by unfolding with topological validation. *Computer-Aided Design*, 39(10):898–913, 2007.

[41] Yang Liu, Jungwook Park, R.J. Lang, A. Emami-Neyestanak, S. Pellegrino, M.S. Humayun, and Yu-Chong Tai. Parylene origami structure for intraocular implantation. In *The 17th International Conference on Solid-State Sensors, Actuators and Microsystems*, pages 1549–1552, June 2013.

[42] Heike Matcha and Ante Ljubas. Parametric origami. In *Future Cities: ECAADE 2010: Proceedings of the 28th Conference on Education in Computer Aided Architectural Design in Europe*, pages 243–257, 2010.

[43] Leonid Mirny and Eugene Shakhnovich. Protein folding theory: From lattice to all-atom models. *Annual Review of Biophysics and Biomolecular Structure*, 30(1):361–396, 2001. PMID: 11340064.

[44] Koryo Miura. A note on intrinsic geometry of origami. In *Proceedings of the First International Meeting of Origami Science and Technology*, pages 239–249, 1989.

[45] C. D. Onal, R. J. Wood, and D. Rus. Towards printable robotics: Origami-inspired planar fabrication of three-dimensional mechanisms. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4608–4613, May 2011.

[46] S. A. Robertson. Isometric folding of Riemannian manifolds. *Proc. of the Royal Society of Edinburgh*, 79(3-4):275–284, 1977.

[47] Paul W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.

[48] Thomas J Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226. ACM, 1978.

[49] M. Schenk, S. G. Kerr, A. M. Smyth, and S. D. Guest. Inflatable cylinders for deployable space structures. In *Proceedings of the 1st International Conference Transformables 2013*, pages 18–20, 2013.

[50] Guang Song and N.M. Amato. A motion-planning approach to folding: from paper craft to protein folding. *IEEE Transactions on Robotics and Automation*, 20(1):60–71, Feb 2004.

[51] Tomohiro Tachi. Generalization of rigid foldable quadrilateral mesh origami. In *Symposium of the International Association for Shell and Spatial Structures (50th. 2009. Valencia). Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures: Proceedings.* Editorial Universitat Politècnica de València, 2009.

[52] Tomohiro Tachi. Simulation of rigid origami. *Origami^4: Proceedings of the 4th International Conference on Origami in Science, Mathematics and Education*, pages 175–187, 2009.

[53] Tomohiro Tachi. Origamizing polyhedral surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):298–311, 2010.

[54] Tomohiro Tachi. Rigid-foldable thick origami. *Origami^5: Proceedings of the 5th International Conference on Origami in Science, Mathematics and Education (OSME 2010)*, pages 253–264, 2011.

[55] Tomohiro Tachi. Freeform origami. Software., 2014. URL `http://www.tsg.ne.jp/TT/software/`.

[56] Tomohiro Tachi and Gregory Epps. Designing one-dof mechanisms for architecture by rationalizing curved folding. In *International Symposium on Algorithmic Design for Architecture and Urban Design (ALGODE-AIJ)*, 2011.

[57] A.P. Thrall and C.P. Quaglia. Accordion shelters: A historical review of origami-like deployable shelters developed by the {US} military. *Engineering Structures*, 59(0):686–692, 2014.

[58] Martin Trautz and Arne Kunstler. Deployable folded plate structures-folding patterns based on 4-fold-mechanism using stiff plates. In *50th. Symposium of the International Association for Shell and Spatial Structures (2009. Valencia). Evolution and Trends in Design, Analysis and Construction of Shell and Spatial Structures: Proceedings.* Editorial Universitat Politècnica de València, 2010.

[59] Takuya Umesato, Toshiki Saitoh, Ryuhei Uehara, and Hiro Ito. Complexity of the stamp folding problem. In *Combinatorial Optimization and Applications*, pages 311–321. Springer, 2011.

[60] F. Wang and L. Chang. Determination of the bending sequence in progressive die design. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 209(1):67–73, 1995.

[61] Wei Yao and Jian S Dai. Dexterous manipulation of origami cartons with robotic fingers based on the interactive configuration space. *Journal of Mechanical Design*, 130(2):022303, 2008.

[62] Shannon A. Zirbel, Robert J. Lang, Mark W. Thomson, Deborah A. Sigel, Phillip E. Walkemeyer, Brian P. Trease, Spencer P. Magleby, and Larry L. Howell. Accommodating thickness in origami-based deployable arrays. *Journal of Mechanical Design*, 135(11):111005–111016, 2013.