How to cut shapes

Shintaro Fushida-Hardy

December 1, 2019

Abstract

Given any polygon drawn on a piece of paper, possibly with holes, can the paper be folded in such a way that cutting it once in a straight line and then unfolding it produces the desired shape? This is the *fold and cut problem*, which was solved around 20 years ago by Erik Demaine, Martin Demaine, and Anna Lubiw. We will interactively explore two solutions to this problem. Join me for a fun afternoon of chopping up paper! (Paper and scissors will be provided.)

Disclaimer: the diagrams in this document are terrible, since I was too busy to make pretty pictures.

1 Motivation

KIDDIE Colloquium talk: *Krafting In 2-D, I.E. folding and cutting colloquium.* Often in an arts and crafts setting, one will want to cut shapes out of a piece of paper. The standard approach is to cut into the paper from the side, destroying the "outside" of the shape, but obtaining a clean copy of the desired shape.

Demonstration: cutting a circle out of a piece of paper.

On other occasions, one might wish to make a circular hole in a piece of paper. A common approach is to pierce the paper in a region which is going to be removed anyway. Then you cut towards an edge and cut out the circle. This time the "outside" of the circle looks good, but the circle itself has been destroyed.

Demonstration: cutting circular hole in a piece of paper.

So what should I do if I want to cut a circular hole in a piece of paper in such a way that both the inside and outside of the circle are preserved?

Demonstration: fold paper in half, then cut.

Scissors are useful because we can cut along curved paths, but maybe you want to cut out some shapes and the only tool you have is a guillotine. Then no matter how many times you fold the paper, you'll never be able to cut out a circle. More generally, you can't cut any shape that has curved edges. This begs the following question: which shapes, (comprised of straight edges), can we cut out from a piece of paper by folding it up and slicing it under a guillotine?

Demonstration: fold paper, cut out square with one cut.

I now have some tasks for everyone, in approximately increasing difficulty as follows:

Exercise 1.1. *1. Fold a piece of paper and make a single straight cut so that it unfolds to a regular n-gon (for some n).*

2. Fold a piece of paper and make a single straight cut so that it unfolds to a star. (Six-pointed star is probably easiest.)

3. Fold a piece of paper and make a single straight cut so that unfolds to a shape that isn't connected. For example, two square holes side by side.

4. Fold a piece of paper and make a single straight cut so that it unfolds to a shape which isn't simply connected. For example, a square with a square hole inside it. (Can you generalise this to cut out shapes with arbitrarily many holes?)

Demonstration: fold paper, cut out 3-holed annulus with one cut.

2 1-skeleton method

Loosely speaking, this shows that we can make shapes with lots of edges, which aren't necessarily connected or simply connected. This leads to the following conjecture:

Any shape comprised of finitely many piece-wise linear edges (not necessarily connected or simply connected) can be cut out from a piece of paper.

I suspect that everyone has noticed that all of the solutions so far heavily rely on symmetry. Even though we've established that topologically arbitrarily complex shapes can be created, we're very far from knowing how to cut out asymmetric things.

Exercise 2.1. *1. Fold a piece of paper and make a single cut so that it unfolds to a given arbitrary (asymmetric) triangle.*

Demonstration: fold paper, cut out arbitrary triangle.

In fact, this gives us a general approach that might work on all "generalised polygons".

(a) Determine 1-skeleton of the polygon.

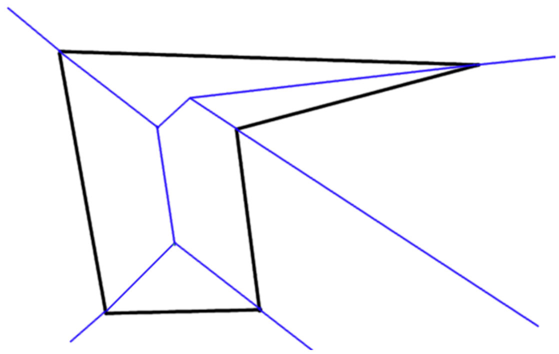
(b) Draw all edge-orthogonals, reflecting through the 1-skeleton where necessary.

(c) Fold along these lines!

The best way to explain this procedure is through an example. Can anyone give me an example of a not-too-complicated but slightly non-trivial shape?

(a) 1-skeleton: Draw the shape. Consider the time-evolution of the shape, given by expanding/contracting the shape with a constant orthogonal speed. The path traced by vertices determines the 1-skeleton.

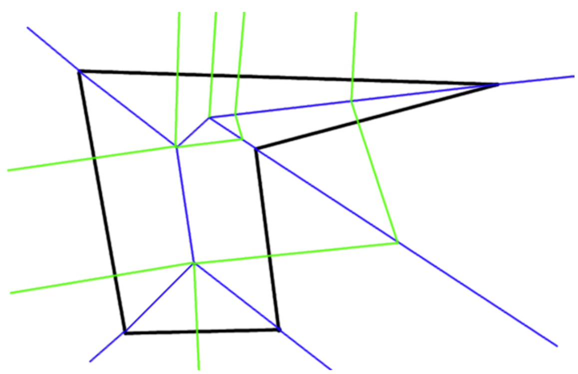
Demonstration: draw example on board.



(b) egde-orthogonals: The 1-skeleton is intuitively useful, because they correspond to angle bisectors. This means they work to bring all of the edges of our polygon onto each other. However, it isn't enough: there is no way to fold along these lines in general. To remedy this, we add more crease-guides. Intuitively the locations where "folding the paper into something flat" fails are the vertices of the 1-skeleton. Drawing lines from the 1-skeleton which are orthogonal to the polygon edges corresponds to adding more angle bisectors (as the orthogonal bisects the angle π).

Unfortunately the orthogonals can't always be drawn in a naive way, as the 1-skeleton can get in the way. When this happens, to ensure that our shape folds properly, we reflect through the 1-skeleton.

Demonstration: drawing orthogonals.



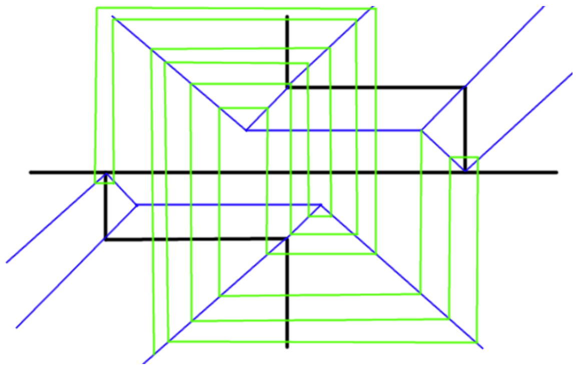
To determine whether or not this solves the problem, we need to verify two things:

(i) The above procedure always gives finitely many crease-guides.

(ii) There exists a flat folding of the paper along creases determined by only the 1-skeleton and orthogonals.

It turns out that this doesn't work! (i) can fail.

Demonstration: crease-guides not necessarily finite.



Conjecture 2.2. *Given a random graph, the orthogonals of its 1-skeleton are dense with probability 0.*

If the conjecture is true, then tiny perturbations of our desired polygon should always be fold-and-cut-able. However, the 1-skeleton method will not work in full generality.

Exercise 2.3. *1. Use the 1-skeleton method to determine a folding of some shape that you like.*

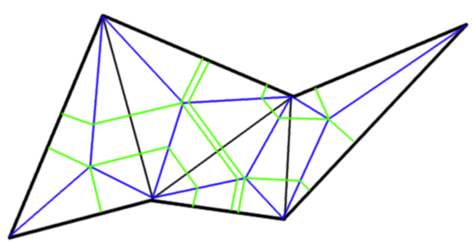
3 Disk-packing method

Although the 1-skeleton approach doesn't actually work in full generality, there is a slightly more impractical algorithm which is guaranteed to work. In the remainder of this talk we provide a method of folding and cutting any arbitrary polygon!

Earlier we observed that given an arbitrary triangle, there is a simple deterministic way of folding it so that a single cut gives the desired shape. Instead of generalising the approach of folding the triangle to other shapes, we can generalise in a different direction: turning our desired shape into a collection of triangles.

As an example, I'll begin by showing why any naive decomposition of our shape into triangles does not work in general, and use it to identify an important property that our "triangle decomposition" should satisfy.

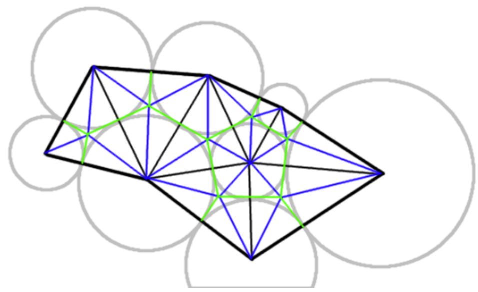
Demonstration: drawing example where orthogonals need to reflect.



If the triangles don't align in a way that forces orthogonals of adjacent triangles to match up, we have no chance: this is just a return to the previous case, where reflecting orthogonals through the other crease lines might never end.

Can anyone think of a shape that might have useful properties to ensure that these orthogonals align?

Demonstration: draw example, with circle packing.

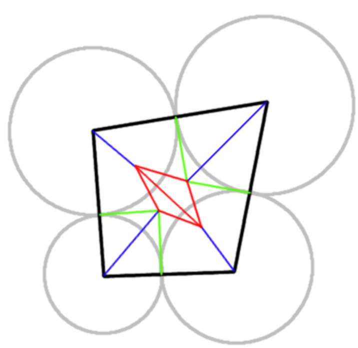


This translates our problem about folding and cutting into a problem about triangle packing, and then into a problem about circle packing. In general we can't pack circles (such that each vertex of the polygon lies at the centre of a circle, and all edges only pass through diameters of circles) so that the cavities are all 3-cavities.

To see this, note that given any three points, there is a unique circle passing through them. Given four points in general position, we typically can't find a circle which will pass through all four points. But if we have a 4-cavity, the only way to turn it into a 3-cavity is by adding a circle which is tangent to four arcs.

To add a bit more freedom, we introduce a method of folding certain quadrilaterals, instead of just triangles. That is, we'll aim for our final "circle packing" will decompose our shape into triangles and quadrilaterals in a way that allows a flat folding.

Demonstration: gusset molecule, courtesy of Robert Lang, my hero. Draw on board first, then also fold it.



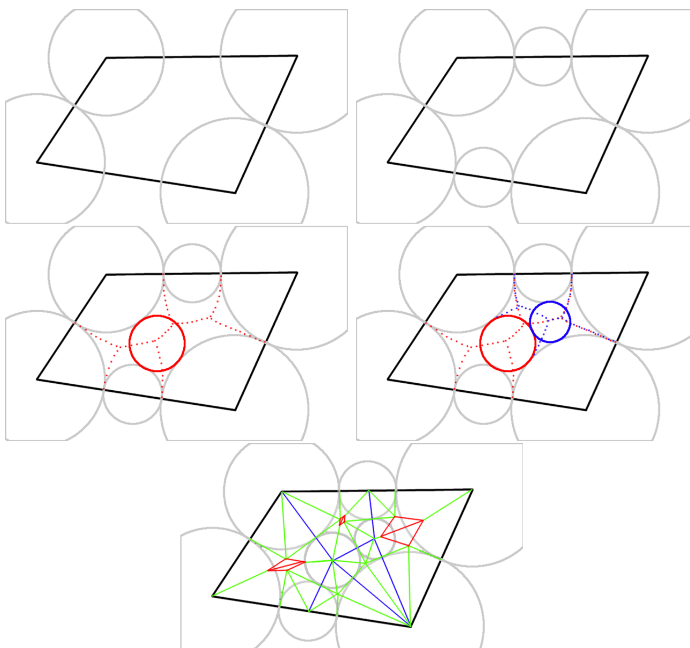
Theorem 3.1. *Given any general polygon, there exists a finite "disk packing" such that all vertices of the polygon lie at centres of disks, and all edges of the polygon pass through diameters of disks.*

Proof. An algorithm is as follows. (One can prove that this terminates in finitely many steps.)

1. Place disks of radius exactly half the distance to the nearest edge/vertex, on every vertex of the polygon.
2. Consider each uncovered edge between vertex disks. If the circle of diameter equal to the uncovered region of the edge has radius at most half the distance to the nearest edge/vertex, draw the circle. Otherwise halve the radius and repeat until it fits.
3. Compute Voronoi diagram of cavities. Add largest radius disk (with origin at a vertex of the Voronoi diagram) that fits inside a cavity. Re-calculate Voronoi diagram, repeat.

□

Demonstration: draw diagrams of this process.



Theorem 3.2. *The circle packing above gives a decomposition of any polygon into "molecules" which are all compatible and provide a method of folding the shape into a flat structure for which one cut suffices.*

Remark. *The outer polygon must be "thickened" to ensure that the interior of the polygon remains intact following the cut.*