Making Polygons by Simple Folds and One Straight Cut

Erik D. Demaine[1], Martin L. Demaine[1], Andrea Hawksley[1], Hiro Ito[2], Po-Ru Loh[1], Shelly Manber[1], and Omari Stephens[1]

[1] MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar St., Cambridge, MA 02139, USA,
{edemaine,mdemaine,hawksley,ploh}@mit.edu,
shellym@alum.mit.edu, xsdg@xsdg.org
[2] School of Informatics, Kyoto University,
Kyoto 606-8501, Japan,
itohiro@i.kyoto-u.ac.jp

Abstract. We give an efficient algorithmic characterization of simple polygons whose edges can be aligned onto a common line, with nothing else on that line, by a sequence of all-layers simple folds. In particular, such alignments enable the cutting out of the polygon and its complement with one complete straight cut. We also show that these makeable polygons include all convex polygons possessing a line of symmetry.

1 Introduction

Take a sheet of paper, fold it flat, and then make one complete straight cut. What shapes can the unfolded pieces have? This *fold-and-cut problem* was introduced formally at JCDCG'98 [3], motivated by a 1922 magic trick by Harry Houdini, but with history going back to a 1721 Japanese puzzle book. The answer is that any pattern of straight-line-segment cuts can be obtained in this way [3,2,6, ch. 17]. More precisely, any graph drawn on a piece of paper with edges as straight line segments can be folded so as to align the vertices and edges of the graph onto a line that contains no other points of paper.

We consider a special case of the fold-and-cut problem, called *simple fold-and-cut*, where we require the folding process to consist of a sequence of all-layers simple folds. Given an existing flat folded state, an *all-layers simple fold* [1] is defined by a line segment that divides the paper into two portions, and consists of folding one of those two portions along the segment, through all layers of paper, by $\pm 180°$, so that afterward all paper is planar. We call a graph *simple-fold-and-cuttable* if a sequence of all-layers simple folds brings the graph's vertices and edges to a line, with no excess paper on that line.

We prove two main theorems.

Theorem 1. *There is a strongly polynomial-time algorithm for determining whether a given (not necessarily convex) simple polygon is simple-fold-and-cuttable, starting from a piece of paper strictly containing the convex hull of the polygon.*

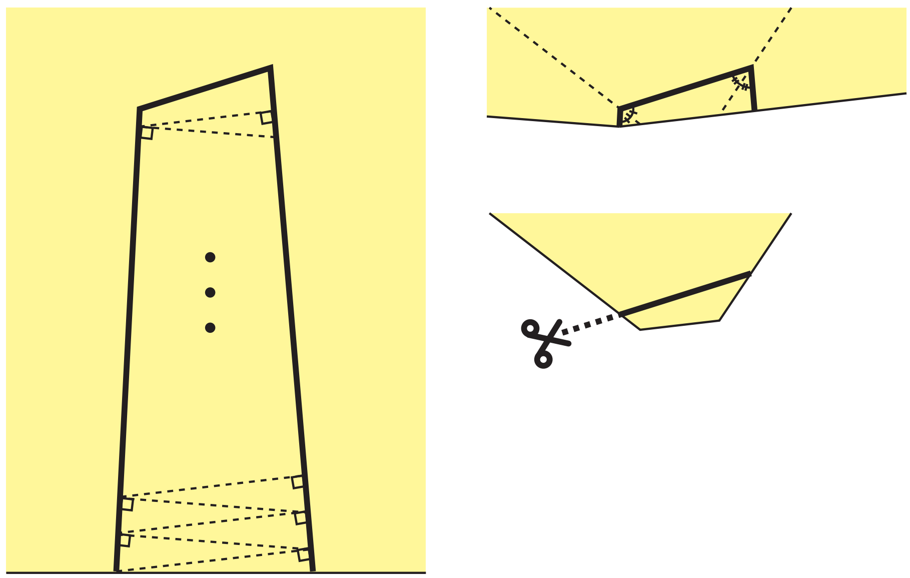


Fig. 1: The graph on the left is simple-fold-and-cuttable, as shown, but as the two acute base angles approach 90°, the number of simple folds grows without bound.

The polynomial running time is a function of the number n of vertices in the input polygon, even though the number of required simple folds can be arbitrary large for a fixed n; see Fig. 1. As a result, when a polygon is simple-fold-and-cuttable, the algorithm produces only an implicit representation of the folding sequence. An explicit representation can be obtained, at the cost of adding to the running time a term linear in the output size.

Theorem 2. *A convex polygon is simple-fold-and-cuttable if and only if it has a line of reflectional symmetry.*

2 Definitions

The input to the (simple-)fold-and-cut problem consists of a *piece of paper* P, which we require to be a polygonal region,[3] and a graph G drawn on P with edges drawn as straight line segments. We view each vertex v of G as a point of P, and thus each edge $\{v, w\}$ of G corresponds to the segment vw, which we require to be contained in P. In fact, we can also view G as a subset of P, namely the set of all points on vertices and edges of G. From this perspective, we naturally remove any degree-2 vertices forming an angle of 180°, as we can

[3] We treat a piece of paper as a closed region, namely, the set of all points interior or on the boundary of the polygon.

simply merge to two incident edges. For convenience, we refer to such a drawing (G, P) as a *graffito*.[4]

We make one assumption about the input graffito (G, P), that the interior of P contains the convex hull of G; we say that (G, P) has a *margin*. This condition is required only of the initial graffito; indeed, it immediately becomes violated after making one fold.

For a graffito (G, P), we call an all-layers simple fold *feasible* if no point of G folds on top of a point in $P \setminus G$; in other words, every point of G folds to either another point of G or a point outside P. Because we consider all-layers simple folds, we can effectively glue together multiple layers of paper into one layer, resulting in a new polygonal region P' of paper, as well as a new graph G' drawn on P'. We write $(G, P) \to (G', P')$ when a feasible all-layers simple fold takes (G, P) to (G', P'). We write $(G, P) \to^* (G', P')$ to denote zero or more transitions $(G, P) \to \cdots \to (G', P')$, allowing in particular $(G, P) = (G', P')$.

A graffito (G, P) is *cuttable* if there is a straight line ℓ such that $G \subset \ell$ (all vertices and edges of G are on ℓ) and $P \setminus G$ is disjoint from ℓ (the rest of the paper is off ℓ). We call a graffito (G, P) *simple-fold-and-cuttable* if $(G, P) \to^* (G', P')$ for some cuttable graffito (G', P').

The main problem considered in this paper is determining whether a given graffito (G, P) with margin, where G is a simple polygon, is simple-fold-and-cuttable. Our model of computation is a real RAM.

3 Passages

The first feasible fold of a graffito (G, P) with margin, where G is a simple polygon, must be a line of reflectional symmetry of the polygon: a simple fold reflects one side onto the other, and cannot map G to anything other than G because of the margin. All such symmetry lines can be found in $O(n)$ time [8].

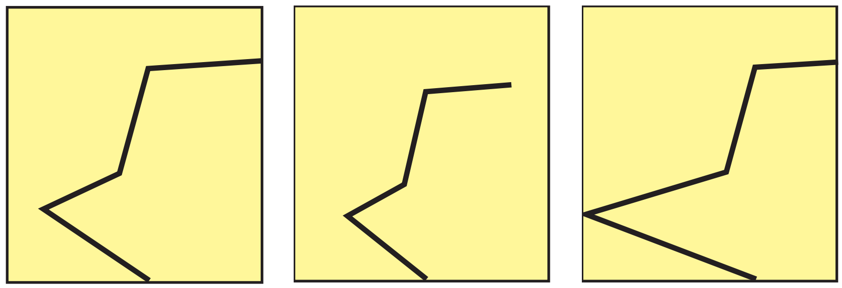
Suppose the first fold has effect $(G, P) \to (G', P')$. Then G' is no longer a polygon, but rather a subchain of G, with endpoints on the boundary of P'. In fact, (G', P') has substantial additional structure, which will be preserved throughout the folding process; we call (G', P') a "passage".

A graffito (G, P) is a *passage* if it satisfies the following four conditions (see Fig. 2):

(a) G is a simple polygonal line,
(b) the two endpoints of G are on the boundary of P,
(c) no point of G except the two endpoints is on the boundary of P, and
(d) for some $\varepsilon > 0$, P contains the *ε-thickened hull* of G: the convex hull of the union of G with ε-radius disks centered at non-end vertices of G.

Properties (b–d) essentially encapsulate what is preserved of the margin property throughout folding, when starting from a simple polygon.

[4] "Graffito" is the singular form of "graffiti" (both in English and Italian), though it seems rarely used in English.



(a) (b) (c)

Fig. 2: (a) is a passage; (b) and (c) are not passages.

4 Algorithm

Our algorithm to determine simple-fold-and-cuttability of a graffito (G, P) with margin, where G is a simple polygon with n vertices, runs in $O(n^2)$ time. The algorithm follows a greedy approach: repeatedly make feasible folds, until either we get stuck or the result is cuttable. This greedy approach is motivated by the following lemma, proved in Section 5.3, which states that feasible folds are always "safe" to make.

Lemma 1. *If (G, P) is a passage, P is convex, and $(G, P) \to (G', P')$, then (G, P) is simple-fold-and-cuttable if and only if (G', P') is simple-fold-and-cuttable.*

If (G, P) is a graffito with margin, G is a simple polygon, and $(G, P) \to (G', P')$, then (G, P) is simple-fold-and-cuttable if and only if (G', P') is simple-fold-and-cuttable.

Proof (sketch). The fold realizing $(G, P) \to (G', P')$ divides G into two parts. Using the properties of passage or margin, we argue that one part is the reflection of a subset of the other part. Thus the first part folds on top of the larger part and effectively disappears; thus $G' \subset G$. (This is the critical point where we use that the original graph is a polygon.) But we may not have $P' \subset P$. To fix this, we fold P' down to $P \cap P'$, which is convex. Now any simple-fold sequence for (G, P) applies as well to (G', P'). □

4.1 High-Level Algorithm

Our algorithm first finds any line of reflectional symmetry of the input polygon, using the $O(n)$-time algorithm of [8], and folds along it. If there is no such line, then the input is trivially not simple-fold-and-cuttable. Otherwise, we obtain a passage, which we denote (G, P).

It remains to characterize simple-fold-and-cuttable passages in $O(n^2)$ time. Initially we mark the endpoints of G as "real"; in the future, the endpoints may become marked as "limit" (meaning that the end edges are in fact slightly longer, but can be shortened arbitrarily close to reaching this endpoint). Throughout, we maintain the invariant that (G, P) is a passage. The algorithm repeatedly loops through the following steps.

Step 1. We replace P with the ε-thickened hull of G for an arbitrarily small (infinitesimal) $\varepsilon > 0$. Here we modify the notion of ε-thickened hull to also include an ε-radius disk for limit endpoints, while real endpoints remain on the hull as before.

Step 2. We look for feasible folds (keeping in mind the infinitesimal extension of limit endpoints and ε-thickened hull) that either hit a vertex of G other than an endpoint, or cross an edge of G other than an end edge. If there is at least one such fold, we arbitrarily chose one and fold it, marking the new endpoint as real.

Step 3. If there are no vertex or non-end-edge folds, but there are feasible folds through end edges, we compute the limit of repeatedly folding folds that cross just the end edges (as detailed below, intuitively mimicking Fig. 1), and mark any modified endpoints as limits.

Exit condition. If there are no feasible folds whatsoever, then we claim that (G, P) is simple-fold-and-cuttable if and only if it is cuttable, i.e., a single edge. Otherwise, the loop proceeds back to Step 1.

4.2 Algorithmic Details

The description above leaves out a few algorithmic details.

Step 1: Computing the convex hull. We can compute the convex hull of the polygonal line G in linear time using, e.g., Melkman's algorithm [7]. The intuition behind replacing P with the ε-thickened convex hull of G is that, as we show in Section 5.4, we can always fold P down to this hull without touching G, and this can create new feasible folds and (by Lemma 1) cannot destroy old feasible folds. Indeed, the ε-thickened convex hull represents the maximum limiting effect we can achieve by making all feasible folds that do not intersect G (while the other steps consider feasible folds intersecting G).

Step 2: Finding feasible folds that hit a non-end vertex/edge. For each non-end vertex and the midpoint of each non-end edge, we test foldability of the angular bisector (treating midpoints as degree-2 vertices) as follows. We start at our vertex or midpoint, choose a direction, and walk along it until we hit the next vertex, say v_1. We check that the fold maps v_1 to a point v_2 on G or outside of P, and also maps an arbitrarily small neighborhood of v_1 onto the corresponding neighborhood of v_2. We must be careful if either v_1 or v_2 is marked as a limit vertex or if some of the margin touching this vertex is infinitesimally small. For example, if v_1 is a limit vertex and v_2 is not, we can consider the fold feasible only if the direction in which v_1 extends folds either to a line segment of G of positive length or folds outside P. Similarly, we cannot assume that P is exactly the convex hull of G: if v_1 is surrounded by an infinitesimally small margin, we must ensure that even this margin folds onto $P \setminus G$ or outside P.

We continue this process until we hit an endpoint of G on either side of our starting vertex. If the reflection of the first endpoint we meet is not also an endpoint, we continue walking from that point away from the fold line, checking to ensure that the image of the remainder of the polygonal line falls outside of P. The running time for each traversal is $O(n)$, and we perform a check for up to $O(n)$ vertices and midpoints, yielding an overall runtime of $O(n^2)$. An interesting open problem is whether this algorithm can be improved to run in $O(n)$ time overall, e.g., along the lines of symmetry finding [8].

Step 3: Computing the limit of folds that cross only end edges. We distinguish two cases according to whether the two rays extending the end edges beyond G's endpoints intersect.

First consider the case in which the rays do not intersect. For each end edge $e = vw$ with end vertex v, we project the union of the non-end edges (which form a path starting at w) onto the line through e, obtaining in $O(n)$ time a line segment s containing w. No fold through e that crosses only end edges can pass through s; thus, if s contains e, then no fold across e is possible. Otherwise, the endpoint of s closer to v is a strict upper bound on the extent to which e can be folded. (Here we use that we have already reduced P to a convex set containing G.)

If the above procedure provides information that some or all of an end edge e' cannot be folded, then we project the nonfoldable segment of e' onto the opposite end edge e and update the bound for e accordingly (see Fig. 3). If we have nonfoldability information about both end edges, then we apply the above step twice, using each edge to update the other.

We show in Section 5.5 that the updated bounds are the desired limits, that is, the end edges can be folded to within infinitesimal amounts of their updated bounds, and in particular, updating the bound for e can give us no new bound on the foldability of e'.

The situation is slightly different if the rays extending the end edges do intersect at a point x: in this case, it is possible that the above approach would cause an infinite cascade of back-and-forth bound updates on the two end edges. Fortunately, in this case, the end edge farther from x is actually not foldable at all (as we will also show in Section 5.5), so it suffices to apply the projection method once to determine the foldability of the end edge closer to x.

5 Correctness

To prove correctness, we first show the invariant that (G, P) remains a passage (Section 5.1). Second we develop tools for folding away excess paper (Section 5.2). Then we show that the exit condition is correct, that is, prove Lemma 1 described

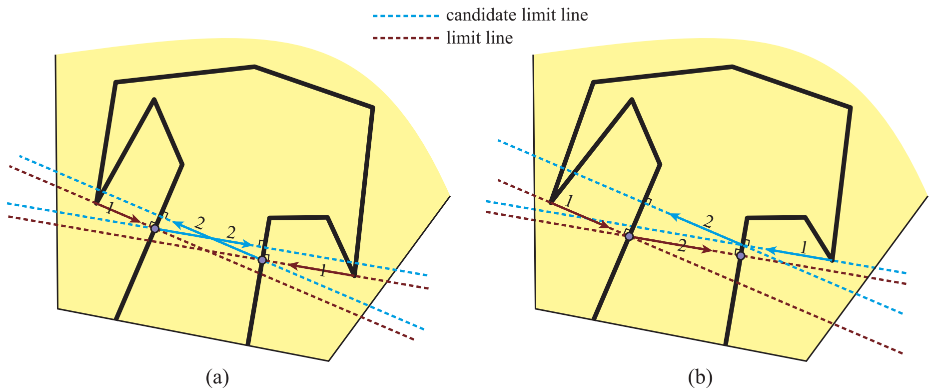


Fig. 3: Computing limit points of end-edge foldability by (1) projecting non-end edges onto end edges and (2) further projecting portions of the end edges determined to be nonfoldable. Limit points computed by the algorithm are identified with circular markers.

above (Section 5.3). Next we show that each step is correct. For Step 1, we need to show how to fold an ε-thickened hull of G, and that this is the limiting effect of folds not intersecting G (Section 5.4). For Step 2, there is nothing to show. For Step 3, we have argued that the limit points we compute are indeed limits, but it remains to show that they are in fact achievable by folding (Section 5.5). Together, these lemmas prove correctness of the algorithm's output, and thus Theorem 1.

5.1 Passage Invariant

Here we show that feasible all-layers simple folds preserve the passage property. In addition, we show that the graph part of the graffito only reduces under such folds, which will be useful in the proof of Lemma 1.

Lemma 2. *Let* (G, P) *be a passage, and suppose that* $(G, P) \to (G', P')$. *Then* (G', P') *is also a passage, and* $G' = G \cap P'$.

The proof is a somewhat tedious geometric/topological argument.

Proof. Refer to Fig. 4. If G does not intersect the fold line segment ℓ realizing $(G, P) \to (G', P')$, then $G = G'$ and clearly (G', P') is still a passage. If G intersects ℓ at exactly one point, say p, and we denote the polygonal lines lying on either side of p by G_1 and G_2, then we have either $G_1 \subseteq G_2$ or $G_2 \subseteq G_1$ in order for the fold to be feasible. Thus G' is the longer of G_1 and G_2, and its endpoints in P' are p and the endpoint of the longer of G_1 and G_2, both of which are on the edge of P'. Because G was a passage, G' satisfies the remaining conditions for being a passage. Clearly $G' = G \cap P'$.

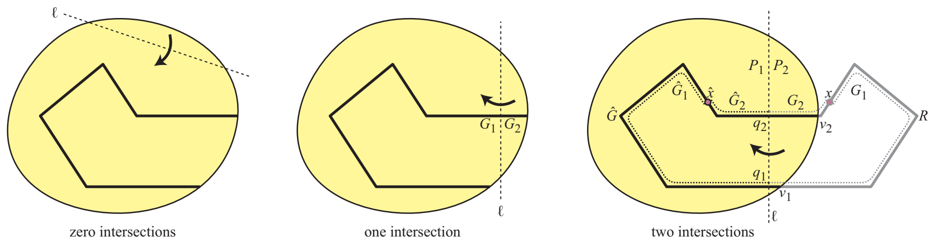


Fig. 4: Proof of Lemma 2.

If G crosses ℓ at two or more points, let $\hat{G}$ be a section of G that connects two consecutive intersection points q_1 and q_2 along ℓ. The line ℓ divides P into two pieces; let P_1 denote the piece containing $\hat{G}$, and let P_2 be the other piece. Because ℓ is a feasible fold, every point of the reflection R of $\hat{G}$ through ℓ is either on G or not in P.

If no point of R lies outside of P, then $R \subseteq G$ and the union of $\hat{G}$ and R is a cycle in G, contradicting that (G, P) is a passage. Thus some point x of R must lie outside of P. Let $\hat{x}$ be the corresponding point of $\hat{G}$ that reflects to x, and consider the polygonal segment $\hat{G}_1$ connecting q_1 to $\hat{x}$. Because the reflection of q_1 through ℓ is q_1, which is in P, and the reflection of $\hat{x}$ is x, which is not in P, the reflection G_1 of $\hat{G}_1$ intersects the edge of P and thus contains an endpoint of G, say v_1. Similarly, if $\hat{G}_2$ is the path connecting q_2 and $\hat{x}$, then its reflection G_2 contains an endpoint v_2 of G.

We cannot have $v_1 = v_2$, for then $v_1 = v_2$ would be connected to both q_1 and q_2 via paths lying only in P_2, hence forming a cycle with G'. So we have identified the two endpoints of G (v_1 and v_2), and have described a polygon consisting of the concatenation of $\hat{G}_1, G_1, G_2, \hat{G}_2$ that enters/exits P only at those two endpoints. Therefore the portion of the polygon interior to P is exactly G itself.

By assumption, G_1, G_2, and $\hat{G}$ do not intersect ℓ outside of q_1 and q_2. Thus G crosses ℓ at exactly two points. Furthermore, when folded, v_1 and v_2 lie atop $\hat{G}$, which is completely contained in P', and because G was a passage, G_1, G_2, and G' therefore do not touch any edge of P' other than ℓ. Hence $G' = \hat{G} = G \cap P'$ and (G', P') a passage. □

5.2 Shrinking Excess Paper

In two related situations below, we have need for folding the piece of paper P down to a convex subset containing G, via a sequence of feasible folds. This task is similar in spirit to the *hide gadget* of [4], which folds a polygonal piece of paper P down to any desired convex subpolygon, but without respecting such obstacles as those imposed by feasibility and G. Another related construction is the folding of an arbitrary simple polygon down to a small triangular subset, with each fold reducing the polygon of paper to a subset [5,6, Sec. 11.6]. This construction effectively avoids obstacles, because any obstacles must be outside the initial polygon, and all intermediate polygons are within the initial polygon; but the construction does allow specifying a very specific target shape.

We build on the ideas of both of these constructions to obtain the results we need in our context, which in some sense generalize the previous results. Our results proceed in a sequence of increasingly general settings, starting with just a triangle:

Lemma 3. *A given triangle of paper $\triangle abc$, with a specified base edge ab, can be folded down to a trapezoid with base ab whose height and base angles are each at most a specified bound $\varepsilon > 0$. The folding does not disturb the base edge ab, never places material outside $\triangle abc$, and consists of a sequence of all-layers simple folds assuming that edges ac and bc are on the paper boundary.*

The proof of this lemma is similar in spirit to [5, Fig. 6], [6, Fig. 11.11], though the details differ.

Proof. Refer to Fig. 5. If a base angle is obtuse, say at a, then we conceptually split it into two triangles, by cutting along the segment aa' orthogonal to bc. The triangle $aa'c$, with base aa', is non-obtuse. Assuming the lemma for such triangles, we can reduce the triangle to a thin triangle against aa', at which point we can fold it into the other triangle aba'. Now we are left with triangle aba' with base ab, which is also non-obtuse. Thus we can apply the lemma again, completing the obtuse case.

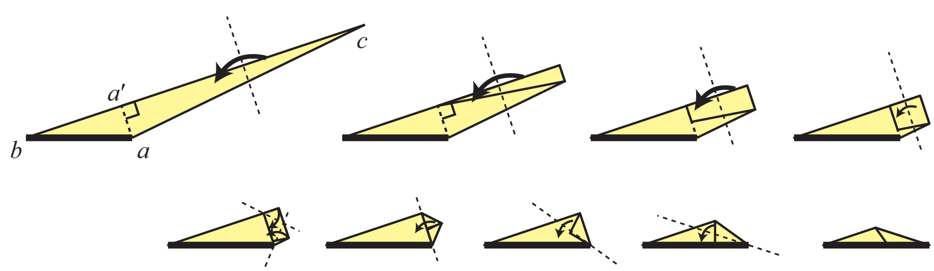


Fig. 5: Shrinking triangle abc down to a thin triangle with common base edge ab. The obtuse case shown in the top left split into non-obtuse cases by cutting along the perpendicular aa'. Then the algorithm repeatedly halves the height until it is possible to repeatedly halve the angles, until the first triangle can be absorbed into the second triangle.

Now suppose that the base angles are non-obtuse. Let c' denote the orthogonal projection of c onto ab. Fold parallel to ab to bring c onto c'. Repeatedly fold along a chord parallel to ab to bring the previous fold onto c', until the height of the resulting trapezoid is at most ε. This process terminates in $O(\log\lceil h/\varepsilon\rceil)$ folds, where h denotes the height of the original triangle.

Having attained the height bound, it remains to attain the angle bounds. Fold alternately along the angular bisector at a and along the angular bisector at b, until the angles are at most ε. Because the angles at a and b are non-obtuse, each such pair of folds does not move material outside the original trapezoid, provided ε was sufficiently small. (If the fold would go outside the trapezoid because the height is too large, we retroactively reduce ε for the height reduction step.) Thus the fold preserves that the piece of paper is a trapezoid, possibly degenerating to a triangle (in which case the folds decrease the height in addition to angles). This process terminates in $O(\log\lceil\theta/\varepsilon\rceil)$ folds, where θ is the maximum original angle. □

Lemma 4. *A given simple polygon of paper with specified base edge can be folded down to a trapezoid with the same base whose height and base angles are each at most a specified bound $\varepsilon > 0$. The folding does not disturb the base edge, never places material outside the original polygon, and consists of a sequence of all-layers simple folds assuming that all nonbase edges of the polygon are on the paper boundary.*

The proof of this lemma uses similar ideas to [5, Lem. 1], [6, Lem. 11.6.1].

Proof. Triangulate the polygon (without Steiner points). Every polygon triangulation has at least two ears (triangles with two boundary edges), so has at least one ear T that is not the triangle incident to the base edge. Apply Lemma 3 to T, considering its one nonboundary edge e as its base. Now fold the resulting trapezoid along e. By setting ε smaller than all angles and orthogonal distances in the original triangulation, this trapezoid will fold within the other triangle sharing edge e. Thus we obtain a triangulation with one fewer triangle. By induction, we reduce to the base case that the polygon is a triangle, which is handled by Lemma 3. □

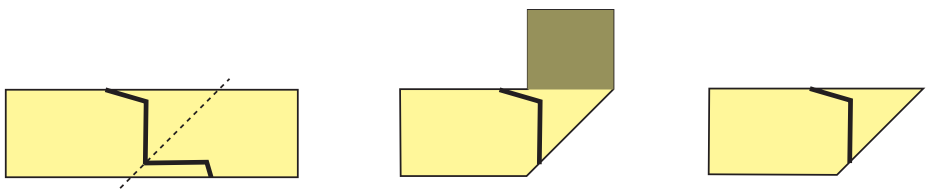
Lemma 5. *A given simple polygon of paper, with a specified reflex chain R, can be folded to form the region between R and a convex chain C with the same endpoints, such that every point of C is within distance ε of a point on R and the shared endpoints of R and C form an angle of at most ε, for a specified bound $\varepsilon > 0$. The folding does not disturb the reflex chain, never places material outside the original polygon, and consists of a sequence of all-layers simple folds assuming that all edges of the polygon not on R are on the paper boundary.*

Proof. Let $r_1, r_2, \ldots, r_n$ denote the sequence of edges along chain R. Let e_1 denote the extension of r_1 in both directions until it hits the boundary of the polygon of paper P, which is a portion of a line of support of R. Apply Lemma 4 to the portion of P on the side of e_1 not containing R, with e_1 as the specified base edge. Repeat this process for r_2, r_3, …, r_n: extend r_i to form e_i, and apply Lemma 4 to the side of P opposite R. The resulting piece of paper is a union of trapezoids output by Lemma 4, which implies all the desired properties. □

5.3 Feasible Folds are Safe

This section demonstrates that, after performing any feasible fold, we can preserve the convexity of the piece of paper by folding away any extra, as shown in Fig. 6. It then concludes with the proof of Lemma 1.

Lemma 6. *Let (G, P) be a passage with P convex, and suppose $(G, P) \to (G', P')$. Then there is a convex piece of paper $P'' \subset P$ for which $(G', P') \to^* (G', P'')$ and (G', P'') is a passage.*



(a) (G, P) (b) (G', P') (c) (G', P'')

Fig. 6: (a) (G, P) has a convex piece of paper P, but one fold leads to (b) (G', P') with nonconvex paper P' that is partly exterior to P. The shaded area $P' - P$ can be folded away to produce (c) (G', P'') with P'' convex and contained in P.

Proof. Let ℓ be the fold line realizing $(G, P) \to (G', P')$. By Lemma 2, we know that $G' \subseteq G$ and that (G', P') is a passage. Because P is convex, line ℓ divides P into two convex pieces, say P_1 and P_2. Because G' lies wholly on one side of ℓ, and $G' \subseteq G \subset P$, we must have G' wholly contained in either P_1 or P_2. Let P'' denote the P_i containing G', and let P_0 be the other (P_{3-i}). Thus $G' = G \cap P''$. Because (G', P') is a passage, G' cannot touch the boundary of P'' except possibly at its endpoints. Because (G, P) is a passage, $G' \subseteq G$ cannot have an endpoint interior to the reflection R of P_0 through ℓ, for such an endpoint would reflect to an endpoint of G interior to $P_0 \subset P$. Therefore (the endpoints of) G' cannot touch the boundary of P'' interior to R.

Now we apply Lemma 5 to each connected component of $P' \setminus P''$, with the reflex chain corresponding to the shared boundary with P'', and a sufficiently small ε. Then we repeatedly fold along edges of the reflex chains, to place the narrow bands from Lemma 5 inside P''. Because the angles at the ends of the reflex chains are less than ε, these foldings will not hit G' even if G' touches the endpoints of the chain. (As argued above, G' cannot touch the interior of a chain.) The convergence of the foldings in a finite number of folds is nontrivial if there are acute angles along the reflex chain, because one fold may "pollute" the adjacent part of the band. Fortunately, this argument is essentially identical to an existing argument for the second half of the hide gadget [4, Theorem 2].

Therefore we obtain a sequence of all-layers simple folds from (G', P') into (G', P''). By construction, P'' is convex and $P'' \subset P$, as desired. $\square$

Proof (of Lemma 1). If (G', P') is simple-fold-and-cuttable and $(G, P) \to (G', P')$, then preceding the sequence of folds to cut (G', P') by the fold $(G, P) \to (G', P')$ yields a sequence of folds to cut (G, P), so (G, P) is simple-fold-and-cuttable.

Conversely, if (G, P) is a simple-fold-and-cuttable passage and $(G, P) \to (G', P')$, then Lemma 6 shows that there is a convex piece of paper $P'' \subset P$ for which $(G', P') \to^* (G', P'')$ and (G', P'') is a passage. Lemma 2 further shows that $G' = G \cap P'$. Because $G' \subset P'' \subset P'$, we have $G' = G \cap P''$. This means that by considering (G', P'') as a subset of the graffito (G, P), and applying the sequence of folds needed to cut (G, P), we will have folded (G', P'') into a cuttable diagram as well. Thus, preceding this sequence of folds with the sequence $(G', P') \to^* (G', P'')$ gives a sequence to fold and cut (G', P').

Finally, if (G, P) is a simple-fold-and-cuttable graffito with margin, where G a simple polygon, and $(G, P) \to (G', P')$, then we follow a similar argument. Because the fold is a line of symmetry, we know that $G' = G \cap P'$. By the margin property and symmetry, G' must be contained in the interior of $P \cap P'$, except along the fold line segment ℓ. Therefore we can reduce P' to $P'' = P \cap P'$ simply by applying Lemma 4 to each connected component of $P' \setminus P$, then folding the small triangles along their bases to absorb them into P''. Thus, as before, we obtain $(G', P') \to^* (G', P'')$ where (G', P'') can be folded in the same ways as (G, P). □

5.4 Approaching the Convex Hull (Step 1)

Because of the invariant that P contains the convex hull of P (the fourth property of passage), any feasible fold that does not intersect G also cannot intersect the convex hull of G, even on its boundary. Thus such folds must in fact be at least some infinitesimal distance away from the hull, which is our ε. Therefore such folds can never reduce P below an ε-enlarged hull of G, for some $\varepsilon > 0$. It remains to show that we can actually achieve the ε-enlarged hull of G, for any desired $\varepsilon > 0$.

Lemma 7. *Any passage (G, P) can be simply folded to the passage (G, P') where P' is the ε-enlarged hull of G, for any specified $\varepsilon > 0$ sufficiently small that $P' \subseteq P$.*

Proof. The polygonal line G (plus short extensions at limit endpoints) divides the ε-enlarged hull P', as well as the piece of paper P, into two halves. We process each half separately, applying Lemma 5 to each connected component of $P \setminus P'$, where the reflex chain is the shared boundary with P'. Now we repeatedly fold along the line extending each of these shared boundary edges, to place the narrow bands within P'. Because of the small distance between convex and reflex chains, these foldings will not cause paper to fold atop edges of G; and because of the small end angles, these foldings will not cause paper to fold atop endpoints of G. As in the proof of Lemma 6, this process is similar to the second half of the hide gadget [4, Theorem 2] and converges after finitely many folds. □

5.5 Approaching Limit Endpoints (Step 3)

Lemma 8. *Any passage (G, P) can be simply folded to reduce the end edges of G down to ε larger than the limits computed in Step 3.*

Proof. The proof requires casework involving analysis of various configurations, so we begin by proving the lemma in two easy but general situations that will allow us to quickly dispose of several configurations.

First we show that, if the dot product of the end edges (viewed as vectors pointing in the directions of G's endpoints) is nonpositive, then our limit computation is correct. In this case, the foldability of the two end edges is independent in the sense that the shortening of one end edge does not affect the foldability of the other. This independence holds because the projection of one end edge e_1 onto the other edge e_2 yields a segment of e_2 whose endpoint closest to the edge is the projection of that point of e_1 lying farthest from the edge. Thus no shortening of e_1 will improve that bound on foldability of e_2 given by the projection. Having established independence of the end edges, the same argument applies to each end edge.

Next, we show

(∗) If our limit computation correctly determines that one end edge cannot fold at all, then it returns the correct answer for the other end edge e.

Indeed, in this case our algorithm projects all edges other than e onto the line through e, giving a clear upper bound on end-edge-only foldability, and we can fold arbitrarily close to this bound by successively halving the paper beyond it. Note that the halving technique ensures that no paper ever folds past the perpendicular to e at the bound, so these folds are indeed valid: neither the paper being folded away nor the paper being folded onto can contain any points of G other than those on e.

We are left to consider cases in which the end edge vectors form an acute angle with each other. We split into cases based on the locations of the endpoints of G relative to the intersection x of the lines through the end edges, with one additional case for parallel end edges. For convenience, assume the diagram is oriented such that the angle bisector of the end edges is vertical; see Fig. 7.

Case 1: Both endpoints below x. (Note that it is possible that one of the end edges could start above x—unlike in the diagram—but the argument works whether or not this happens.) It is clear that the limit points our algorithm identifies are upper bounds on foldability, so we are left to prove that these limits are attainable. The argument has three main parts:

(a) For each end edge, draw its perpendicular through its limit point; call these perpendiculars *limit lines.* Then all non-end edges are above both limit lines (by definition), and each limit line intersects the opposite end edge either at its limit point or at a point farther along the end edge vector. The latter property follows from the assumption that the end edge vectors are diverging from x with a little geometric reasoning; see Fig. 3.

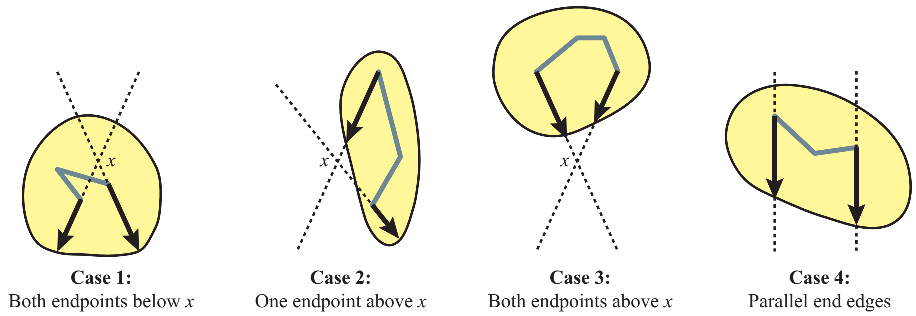


Fig. 7: Illustrations for the proof of Lemma 8.

(b) Ignoring the region above the limit lines for the moment (which as we saw contains all non-end edges), alternately folding the end edges (as in Fig. 1) reduces the endpoints toward x in a geometric series. In particular, it is possible to reach the limit points after a finite number of folds and (importantly) without needing to reduce either end edge beyond its limit point, using the property in (a) about limit line intersection.

(c) Although the folds in the geometric progression in (b) might not actually be feasible because of conflicts with non-end edges, we can overcome this issue by replacing each fold with a series of successive-halving folds that never touch the region above the limit lines.

Case 2: One endpoint above x. First, the tail of the lower end edge must be below x (as depicted) by convexity, because the head of the upper end edge is at the boundary of the paper. It follows that the projection of the non-end edges onto the upper end edge will completely cover the upper end edge, allowing our algorithm to deduce that the upper end edge is not foldable at all. Thus, we are in situation $(*)$ which we have already solved.

Case 3: Both endpoints above x. This case our algorithm treats separately, claiming that the end edge farther from x does not fold at all. We need to show that this determination is correct, after which the correctness of the limit computation for the other end edge follows by situation $(*)$. In fact, *neither* end edge can fold past the distance from the left (short) end edge to x: doing so would require the opposite edge to have already folded beyond that point (because the end edge vectors are angled toward each other), a contradiction.

Case 4: Parallel end edges. This case is essentially one-dimensional, and the argument used for situation $(*)$ applies. □

6 Convex Polygons

We begin by introducing one piece of terminology that will be of use in proving Theorem 2. Let (G, P) be a passage with end edges $e_1 = v_1w_1$ and $e_2 = v_2w_2$

such that v_1 and v_2 are the endpoints of G. We say that (G, P) is *half-convex* if the union of G with segment v_1v_2 is a convex polygon and the total turn angle of the chain of edges from e_1 to e_2 is at most 180°.

Proof (of Theorem 2). As explained in Section 4, a convex polygon with no line of symmetry does not admit an initial fold; thus, it remains to show that a convex polygon with a line of symmetry is simple-fold-and-cuttable.

Begin by folding along any line of symmetry of the polygon, leaving a half-convex passage (G, P). Note that half-convexity implies that (using the notation above) the extensions of e_1 and e_2 backward through w_1 and w_2 meet at a point x; in the case that e_1 and e_2 are parallel, x is understood to be a point at infinity. Also note that all points of G lie within the (non-reflex) angle A between rays xv_1 and xv_2.

For convenience, orient the paper such that the angle bisector of A points vertically downward. Assume without loss of generality that w_1 is further down than w_2. We will show that e_1 and e_2 can be folded small enough that vertex-folding through w_1 is feasible. By making this fold, we will be left with a passage with one fewer edge, and it is easy to see that this passage is still half-convex. By induction on the number of edges, we may then conclude that any half-convex passage is simple-fold-and-cuttable.

We first claim that the limit point on e_1 computed in Section 4.2 is w_1: that is, neither the projection of all non-end edges onto e_1, nor their projection onto e_2 followed by e_1, covers w_1. To see this, simply observe that these projections can only move points inside angle A upward. Because w_1 is (by assumption) the furthest down of all points on non-end edges and G lies inside angle A, the claim follows.

Next, we claim that the limit point u_2 on e_2 is either w_2 or the projection of w_1 onto e_2 (if this projection lies within e_2). Indeed, the same argument as above implies that u_2 is the lower endpoint of the projection of the non-end edges onto e_2 (i.e., we need not bother projecting onto e_1 and back onto e_2), and since the chain of non-end edges lies within triangle xw_1w_2 by convexity, it follows that this lower endpoint must be the result of projecting either w_1 or (trivially) w_2.

Now Lemma 8 shows that we can feasibly fold (G, P) to make e_1 arbitrarily small and reduce e_2 arbitrarily close to u_2. In particular, we can fold e_1 to be shorter than its adjacent edge. Since the angle between e_1 and its adjacent edge is strictly less than 180°, the angle bisector, if it hits G at all, must hit e_2 below u_2 by some discrete amount. So by folding e_2 close enough to u_2, we can feasibly fold along the angle bisector through w_1. □

7 Conclusion

Perhaps the most interesting open question is to characterize simple-fold-and-cuttable graffiti (G, P) beyond when G is a polygon. Fig. 8 shows that, in this case, not every feasible fold is safe.

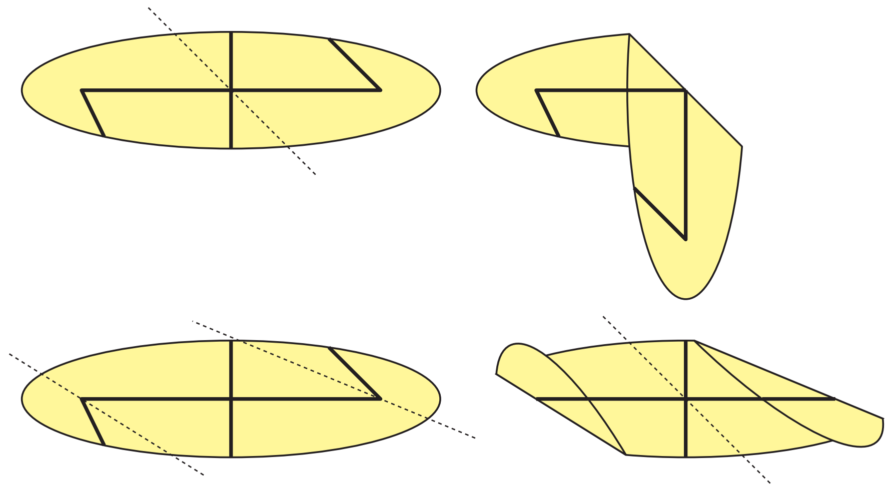


Fig. 8: Allowing a degree-4 vertex, the top fold leads to an unfoldable graffito, while the bottom folds lead to a simple-fold-and-cuttable graffito.

We conjecture that our algorithm works if we relax the margin constraint to allow graffiti (G, P) where G is a polygon whose vertices might lie on the boundary of P (but we still forbid edges of G from lying along P). This situation naturally leads to a relaxed notion of passages where the polygonal line's vertices may lie on the boundary of P. Our greedy algorithm is in fact incorrect for such generalized passages; see Fig. 9. We conjecture that such counterexamples cannot arise from polygons G.

References

1. Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Martin L. Demaine, Joseph S. B. Mitchell, Saurabh Sethia, and Steven S. Skiena. When can you fold a map? *Computational Geometry: Theory and Applications*, 29(1):23–46, September 2004.
2. Marshall Bern, Erik Demaine, David Eppstein, and Barry Hayes. A disk-packing algorithm for an origami magic trick. In *Origami^3: Proceedings of the 3rd International Meeting of Origami Science, Math, and Education*, pages 17–28, Monterey, California, March 2001. Improvement of version appearing in *Proceedings of the International Conference on Fun with Algorithms*, Isola d'Elba, Italy, June 1998, pages 32–42.
3. Erik D. Demaine, Martin L. Demaine, and Anna Lubiw. Folding and cutting paper. In J. Akiyama, M. Kano, and M. Urabe, editors, *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 1763 of *Lecture Notes in Computer Science*, pages 104–117, Tokyo, Japan, December 1998.
4. Erik D. Demaine, Martin L. Demaine, and Joseph S. B. Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. *Computational Geometry: Theory and Applications*, 16(1):3–21, 2000.

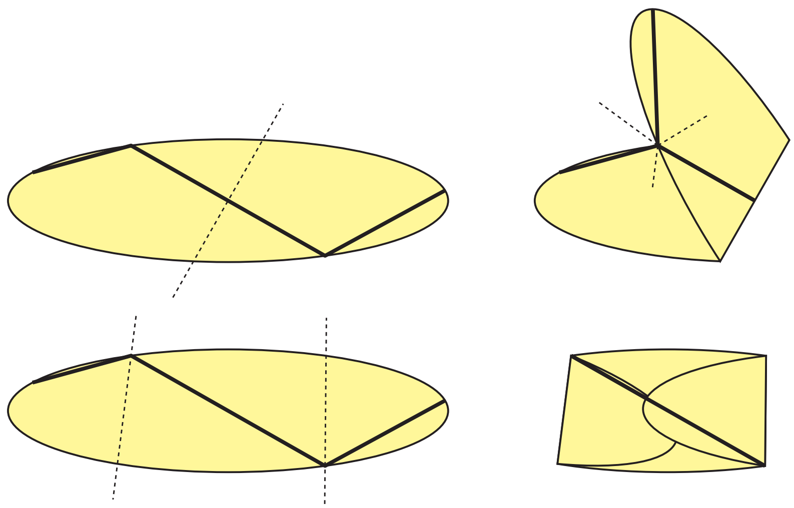


Fig. 9: Our greedy algorithm fails for the "almost passage" on the left: the top fold leads to an impossible-to-fold degree-3 vertex, while the bottom folds result in a cuttable graffito.

5. Erik D. Demaine, Satyan L. Devadoss, Joseph S. B. Mitchell, and Joseph O'Rourke. Continuous foldability of polygonal paper. In *Proceedings of the 16th Canadian Conference on Computational Geometry*, pages 64–67, Montréal, Canada, August 2004.
6. Erik D. Demaine and Joseph O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, July 2007.
7. Avraham A. Melkman. On-line construction of the convex hull of a simple polyline. *Information Processing Letters*, 25(1):11–12, April 1987.
8. Jan D. Wolter, Tony C. Woo, and Richard A. Volz. Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer*, 1(1):37–48, 1985.