Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computational Geometry 31 (2005) 207–218

Computational Geometry
Theory and Applications

www.elsevier.com/locate/comgeo

When can a net fold to a polyhedron? [☆]

Therese Biedl [*], Anna Lubiw, Julie Sun

School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada

Available online 25 January 2005

Communicated by J. Snoeyink

Abstract

In this paper, we study the problem of whether a polyhedron can be obtained from a net by folding along the creases. We show that this problem can be solved in polynomial time if the dihedral angle at each crease is given, and it becomes NP-hard if these angles are unknown. We also study the case when the net has rigid faces that should not intersect during the folding process.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Folding; Net; Polyhedron; Computational geometry

1. Introduction

Folding a polyhedron from a paper pattern has been studied at least since the 15th century, when Albrecht Dürer showed how to fold all regular solids and Archimedian solids [13]. In this paper, we study the case where we are given a *net*, i.e., a polygon and a set of creases, and want to know whether a polyhedron can be obtained by folding along the creases.

We consider two cases, depending on whether we are given the dihedral angle at each crease. If these dihedral angles are given, then we show in Section 3 that the problem can be solved in polynomial time by the simple expedient of performing the folding. If the dihedral angles are not given, then we prove in Section 4 that the problem is NP-hard. We note that an early version of this paper [8] had an incorrect NP-hardness reduction. We then turn to the actual folding process, and give in Section 5 an example of

[☆] These results are part of the third author's Master's Thesis at the University of Waterloo. Research supported by NSERC.
[*] Corresponding author.
E-mail addresses: biedl@uwaterloo.ca (T. Biedl), alubiw@uwaterloo.ca (A. Lubiw).

0925-7721/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comgeo.2004.12.004

a net with rigid faces that can, in the sense above, be folded to form a polyhedron, but only by allowing faces to intersect each other during the folding process.

In the existing literature, a few related problems have been studied. Shephard [20] wrote an early paper on convex nets. Lubiw and O'Rourke showed how to test in $O(n^2)$ time whether an n-vertex polygon with no creases specified can be folded into a convex polyhedron [18]. Another problem is the reverse of ours: Given a polyhedron, can one obtain a net? This can be done for all convex polyhedra [3], as well as for some classes of orthogonal polyhedra [6]. A fundamental open problem in this area is whether for any convex polyhedron there exists a net such that the edges of the net are also edges of the convex polyhedron, a so-called unfolding with edge cuts only. See [14] for some negative results on non-convex polyhedra, and [4] for some positive results for a weaker notion of nets.

Related to our problem of folding a rigid net is the problem of straightening rigid linkages in 3D, which has been studied in [7]. In fact, our proof that some rigid net cannot be folded without intersecting faces is based on the fact that there exists a linkage in 3D that cannot be straightened without having links intersect [7,9].

2. Definitions

A *(polygonal) chain* is a sequence of line segments $[a_i, b_i]$, $i = 0, \dots, n-1$, that are mutually disjoint, except that $b_i = a_{i+1}$ for $i = 0, \dots, n-2$, and possible $b_{n-1} = a_0$. If $b_{n-1} = a_0$, the chain is called *closed*, otherwise it is called *open*. The segments are called *edges* and the endpoints of the edges are called *vertices*. A finite region in the plane bounded by a closed chain is called a *polygon*. A *chord* of a polygon is a line segment inside the polygon where both endpoints are vertices of the polygon. A *net* is a polygon together with a set of chords of the polygon that do not intersect each other except possibly at endpoints. These chords are called the *creases* of the net.

A net can also be viewed as a graph; in fact, it is an embedded *outer-planar graph* since no two edges cross and all vertices are on the unbounded face (the *outer face*). It is known that an n-vertex outer-planar graph has at most $2n-3$ edges. The faces that are not the outer face are called *interior faces*. For an outer-planar graph, the incidences between interior faces form a tree.

In a net, for each crease we may or may not specify the *dihedral angle*, i.e., the angle that the two faces incident to the crease will form inside the finished polyhedron. The dihedral angle cannot be 0 or 2π, because faces are not allowed to overlap. We study two models; in one all folds must actually be folded (so the dihedral angle cannot be π), while in the other there is no such restriction. As it turns out, both models are equal with respect to the complexity of the folding problem.

To be able to test whether a net folds into a polyhedron, we must establish a clear definition of a polyhedron. See [11] for a history of such definitions. We use the following definition, based on Coxeter [10]: A *polyhedron* is a finite connected set of plane polygons, called *faces*, such that (1) if two faces intersect, it is only at a common vertex or a common edge, (2) every edge of every face is an edge of exactly one other face, and (3) the faces surrounding each vertex form a single circuit (to exclude anomalies such as two pyramids with a common apex).

A face of a polyhedron that is parallel to the xy-plane is called an *xy-face*; *xz-faces* and *yz-faces* are defined similarly. An *orthogonal polyhedron* is a polyhedron each of whose faces is an xy-face, xz-face or yz-face.

3. Known dihedral angles

In this section, we show that if dihedral angles are given, we can determine in polynomial time whether folding the net yields a polyhedron. Our model of computation here is the real RAM; for the case of dihedral angles that are multiples of $\pi/2$, basic arithmetic suffices. Our algorithm is straightforward and takes time $O(n^2)$, which is probably not optimal, but suffices to show that the case of unknown orthogonal dihedral angles lies in NP (see Theorem 3).

First, we show how to find the coordinates of the vertices in 3D after the creases have been folded. Compute the tree of adjacencies between the interior faces of the net. Traverse this tree T in depth-first-search order, starting at an arbitrary leaf in an arbitrary position. For each face, the positions of vertices of this face can then be computed using the positions of the vertices of the parent of this face in T, and the dihedral angle that connects the two faces. This takes $O(n)$ time, where n is the number of vertices of the net, because the number of edges and faces of the net is proportional to the number of vertices.

Now we must verify the three properties of polyhedra. We do so in four steps, not for efficiency, but for clarity of presentation: (1) We reject the input if we can find two faces that intersect in a point that is interior to one or both of the faces; (2) We add additional vertices along the edges of the faces, so that any two collinear edges that intersect are equal; (3) For each edge of each face we find all equal edges of other faces, simultaneously building up the *incidence graph*, a data structure to store a polyhedron [16]; (4) We use the incidence graph to test that the faces surrounding each vertex form a circuit.

For step 1 we consider each pair of faces of the net. The running time of our algorithm is then already $\Omega(n^2)$, so we will not worry about making other steps of the algorithm faster than this. Let $F_1, \ldots, F_f$ be the interior faces of the net, and let m_i be the number of edge of F_i. For any $i < j$, if F_i and F_j lie in the same plane, we can test in $O(m_i m_j)$ time how they intersect by testing every pair of edges, and doing a final inclusion test. If F_i and F_j lie in different planes, we compute the line of intersection of these planes and compute the intersections of this line with F_i and F_j, forming two sets of disjoint intervals. We can certainly test in $O(m_i m_j)$ time how these intervals intersect.

The total running time of this step is proportional to

$$\sum_{1\leqslant i<j\leqslant f} m_i m_j \leqslant \frac{1}{2}\left(\sum_{i=1}^{f} m_i\right)\left(\sum_{i=1}^{f} m_i\right) \leqslant \frac{1}{2}(2m)^2,$$

where m is the number of edges of the net. Since the net is an outer-planar graph, $m \leqslant 2n - 3$, so this step takes $O(n^2)$ time.

The addition of extra vertices in the second step is necessitated by the polyhedron property that every edge of every face must be an edge (not just part of an edge) of another face. We need to be careful not to add too many vertices. To implement this step, we gather the edges of faces into collinear groups. Within each group, we sort the endpoints of edges. If there is a point internal to more than two edges in one group, we can reject this input because it violates the second polyhedron property; and if each point is internal to at most two edges then the number of vertices added is linear. The second step can thus be implemented in $O(n^2)$ time, and we end up with $O(n)$ vertices and edges.

In the third step, we test every pair of edges to see whether they are identical. If we ever find an edge that does not have exactly one identical mate, we reject this input. As we match up edges we build the incidence graph that records the incidences of vertices, edges and faces of the polyhedron. This step can be performed in $O(n^2)$ time.

Finally, in the fourth step we use the incidence graph to walk around the faces incident with each vertex and verify that they form a circuit. This can be done in time proportional to the degree of the vertex, and therefore in $O(n)$ time overall.

Our total time therefore is $O(n^2)$. We believe that this time bound can be improved to $O(n \log n)$ by using 3D sweep techniques, rather than the brute-force approach.

4. Unknown dihedral angles

In this section, we show that if dihedral angles are not specified, then the problem of whether a net folds to a polyhedron (from now on referred to as the 3*D Folding Problem*) becomes NP-hard. For our reductions, we use the NP-complete problem PARTITION [17]: Can a set S of positive integers be partitioned into $S = S_1 \cup S_2$, such that the sum of the elements in S_1 equals the sum of the elements in S_2? We calls such a partition a *balanced partition*. We begin with a 2D folding problem that is NP-hard.

4.1. 2D foldings

In 2D, the folding problem is the following: Given a sequence of straight line segments with joints between the segments, determine if we can bend some or all of the joints such that we obtain a polygon, i.e. a closed chain. This problem is solvable in polynomial time, because the answer is positive if and only if no link is longer than all other links together [19]. We study here the *2D orthogonal folding problem*, which is the same problem, except that all joints have to be bent at angles that are a multiple of $\pi/2$. Surprisingly, this makes the problem NP-complete.

Given an orthogonal polygon directed clockwise, the horizontal edges fall into two classes: those directed to the right and those directed to the left; furthermore, the sum of the lengths in each class is the same. This observation is the heart of our reduction of PARTITION to the 2D orthogonal folding problem.

Let $S = \{x_1, \ldots, x_n\}$ be an instance of PARTITION. Set $L = \sum_{i=1}^{n} x_i + 1$, and let $S' = (L, x_1, L, x_2, L, \ldots, L, x_n, L, L, (n+1)L, L)$ be an instance of the 2D folding problem, where the numbers denote the lengths of the links in order along the chain.[1] The sequence formed by deleting the last 3 segments will be called the *jagged sequence*; these segments encode the partition problem. The segments of length L of the jagged sequence will be called the *separation links*. The last three links will be called the *C-links*; these serve to complete the polygon.

Assume the instance S of PARTITION has a solution $S = S_1 \cup S_2$. We construct a solution to the folding problem S' as follows: Working clockwise, the link of length x_i points left if $x_i \in S_1$ and right otherwise, all separation-links point down, and the remaining links form a "C". See also Fig. 1. No two links of the resulting closed chain intersect because from the right ends of the horizontal C-links, we can reach at most $\sum_{x_i \in S_1} x_i = \frac{1}{2} \sum_{i=1}^{n} x_i$ units to the left, so $L > \frac{1}{2} \sum_{i=1}^{n} x_i$ is big enough to prevent any of the links of the jagged sequence from reaching the link of length $(n+1)L$. Observe that no angle is π.

Conversely, assume that S' can be folded into an orthogonal polygon allowing angles of π. Orient the polygon clockwise and define H^+/H^- to be the set of links pointing right/left, and V^+/V^- to be the set of links point up/down. Denote by $|H^+|$ the sum of the lengths of the links belonging to H^+, and

[1] A smaller value of L would suffice for the 2D case, but this value of L is needed for the 3D case.

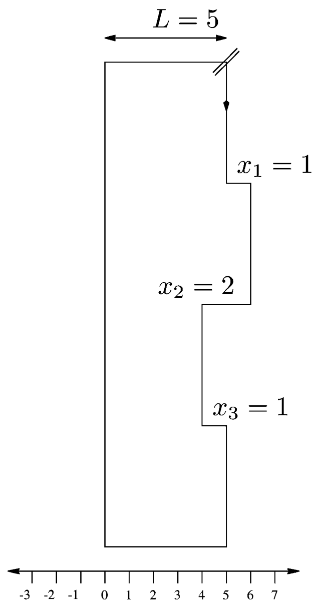

Fig. 1. Constructing S' from the instance $\{1, 2, 1\}$ of PARTITION.

similarly for the other sets. Since we have a closed chain, therefore $|H^+| = |H^-|$ and $|V^+| = |V^-|$, and $|S'| = |H^+| + |H^-| + |V^+| + |V^-| = 2|H^+| + 2|V^+|$. Also,

$$|S'| = (n+1)L + \sum_{i=1}^{n} x_i + L + (n+1)L + L = (2n+5)L - 1 \tag{1}$$

by the definition of L.

Claim. *No two C-links belong to the same set of H^+, H^-, V^+, V^-.*

Proof. Assume two C-links belong to, say, H^+. The third C-link does not belong to H^+, because otherwise $|S'| \geqslant 2|H^+| = 2((n+3)L) = (2n+6)L$, which contradicts Eq. (1). The third C-link also does not belong to H^- by simplicity because it is incident to another C-link. Thus, the third C-link belongs to V^+ or V^-, say to V^+. Then $|S'| = 2|H^+| + 2|V^+| \geqslant 2((n+3)L) = (2n+6)L$, which contradicts Eq. (1). □

Thus, after possible rotation, assume that the three C-links belong to H^-, V^+, H^+ in this order. Thus they form the desired "C".

Claim. *Every separation link belongs to V^-.*

Proof. Assume that one separation link belongs to, say, H^+ (the other cases are similar). Then $|S'| = 2|H^+| + 2|V^+| \geqslant 2(L + L) + 2(n+1)L = (2n+6)L$, which contradicts Eq. (1). Thus the separation links belong to $V^- \cup V^+$. Now $|V^+| \geqslant (n+1)L$, and $|V^-|$ is equal to $|V^+|$ and can only achieve this size if V^- contains all the separation links. □

Finally, by simplicity, no link of length x_i can be in V^+, and thus V^+ consists only of the long C link, and V^- must consist only of the separation links. Thus $H^- \cup H^+$ consists exactly of the links of length x_i, $i = 1, \ldots, n$, and the two C-links of length L, one in each set. Since $|H^-| = |H^+|$, we get the desired balanced partition.

Theorem 1. *The* 2*D orthogonal folding problem is NP-complete, whether angles of π are allowed or not.*

Proof. It is easy to verify that a given assignment of angles of π, $\pi/2$ or $-\pi/2$ forms a polygon, so the problem is in NP. To show that it is NP-hard, assume that an instance S of PARTITION is given and construct the instance S' of the 2D orthogonal folding problem as described above. We have shown that if S has a solution, then S' folds to a polygon where no angle is π. Conversely, we have shown that if S' folds to a polygon, then we can recover a solution to S from the polygon. Hence, PARTITION reduces to 2D orthogonal folding (in both models, i.e., regardless of whether angles of π are allowed), and these problems are both NP-hard. □

4.2. 3D foldings

Now we show that in 3D, both the folding problem and the orthogonal folding problem (where dihedral angles are multiples of $\pi/2$) are NP-hard. The reduction is again from PARTITION, and uses the 2D construction. In our reduction, we construct the net of an orthogonal polyhedron. In order to get NP-hardness of the non-orthogonal 3D folding problem, we show that a balanced partition can be recovered from *any* polyhedron formed by folding this net. We note that the construction in an earlier version of this paper [8] was flawed.

Let $S = \{x_1, \ldots, x_n\}$ be an instance of PARTITION. We describe the construction of an instance of the 3D folding problem in successively more correct refinements. We illustrate these using the PARTITION instance $S = \{1, 2, 1\}$.

The first step is to extrude the polygon formed in Section 4.1 in the z-direction by K units, where K will be specified later. The sequence of links becomes in the net a sequence of rectangles all of height K; see the right hand picture in Fig. 2.

However, the problem then arises of how to cover the front face, see the left hand picture in Fig. 2. Note that the right boundary of the front face abuts the (extruded) jagged sequence, and thus the shape of this face depends upon the partition of S into S_1 and S_2. Even cutting the front face into strips attached to the separation faces as shown in Fig. 2 does not resolve the problem, because the lengths of the strips still depend on the partition.

Our next step is to make the strips that form the front face equally long, which can be achieved by replicating the jagged sequence. We set the width of the faces marked top and bottom to be $L' = (n + 1)L + 2$. The resulting polyhedron looks like a staggered pile of bricks. Its net is independent of any particular partition of S. We also set $K = 2L' + 1$. See Fig. 3 (not drawn to scale).

Unfortunately, this construction is too general: for any input the resulting net folds into a polyhedron. The proof from the two-dimensional case does not transfer because we cannot force the top and the bottom face to be *aligned*, i.e., to have the same xz-projection. See Fig. 4.

To force an alignment of the top and the bottom face, we add an extruded rectangle with x-dimension $\frac{1}{2}$ and y-dimension 1 down the middle of the top face. We call this extruded rectangle the *middle notch*.

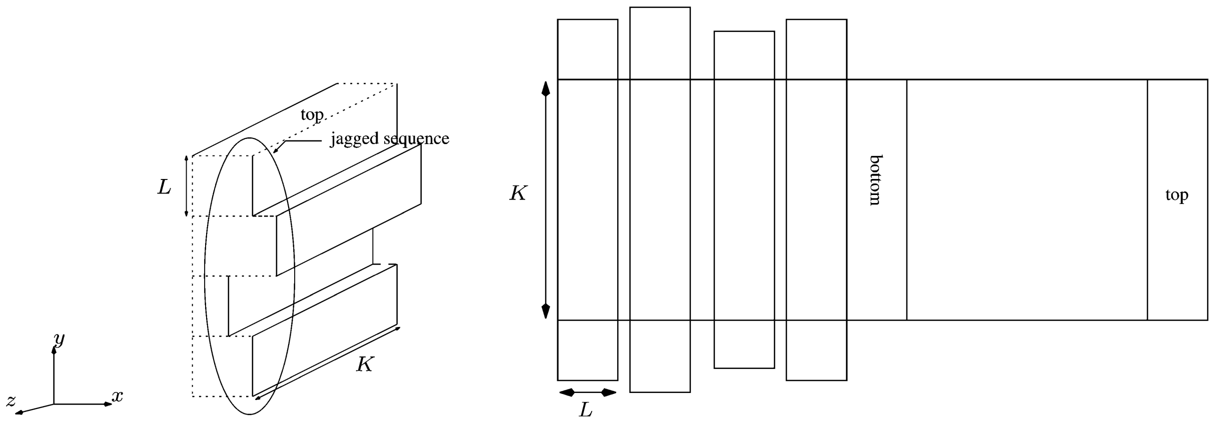

Fig. 2. The construction in 2D extruded and its net. The edges of the polyhedron that result from glueing are dashed. The visible side of the net is the inside of the polyhedron. For clarity, in this and the following pictures, the proportions of the constructed polyhedron and its net are not accurate.

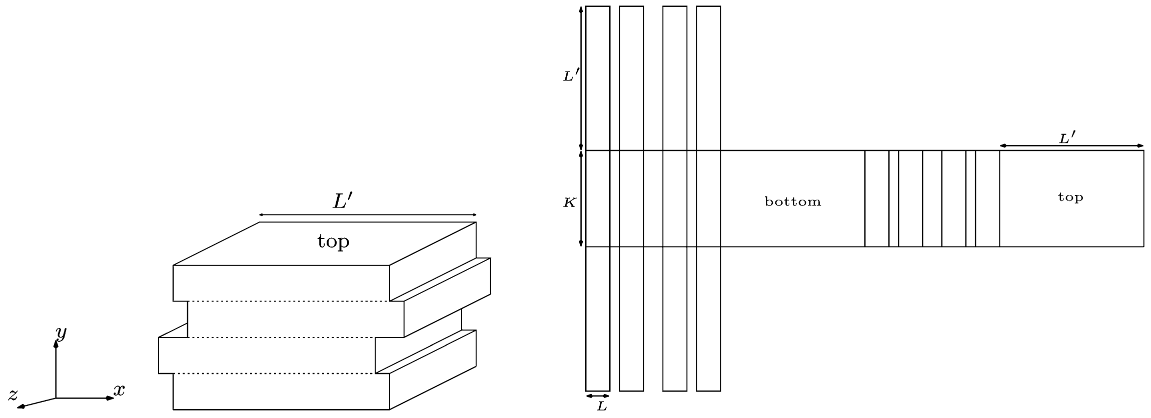

Fig. 3. Replicate the jagged sequence to form a pile of bricks.

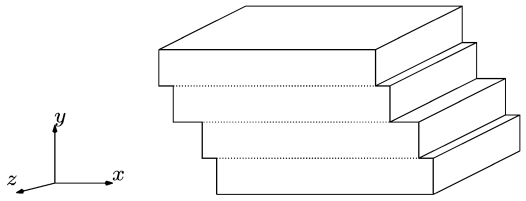

Fig. 4. Unfortunately, now top and bottom face need not align.

We also add a *frame*, attached to the bottom face via another middle notch that will force the two middle notches to be aligned. The frame lies in a rectangle of height $L' = (n+1)L + 2$, and width $2T + 2$ where $T = (n+1)(2L' + L) + 3$; the gap in the top of the frame has width 1, the notch has width $\frac{1}{2}$ and height 1, and the frame has thickness $\frac{1}{2}$. See Fig. 5 (which is not drawn to scale). We call the resulting net S''.

Following the steps of our construction, we obtain:

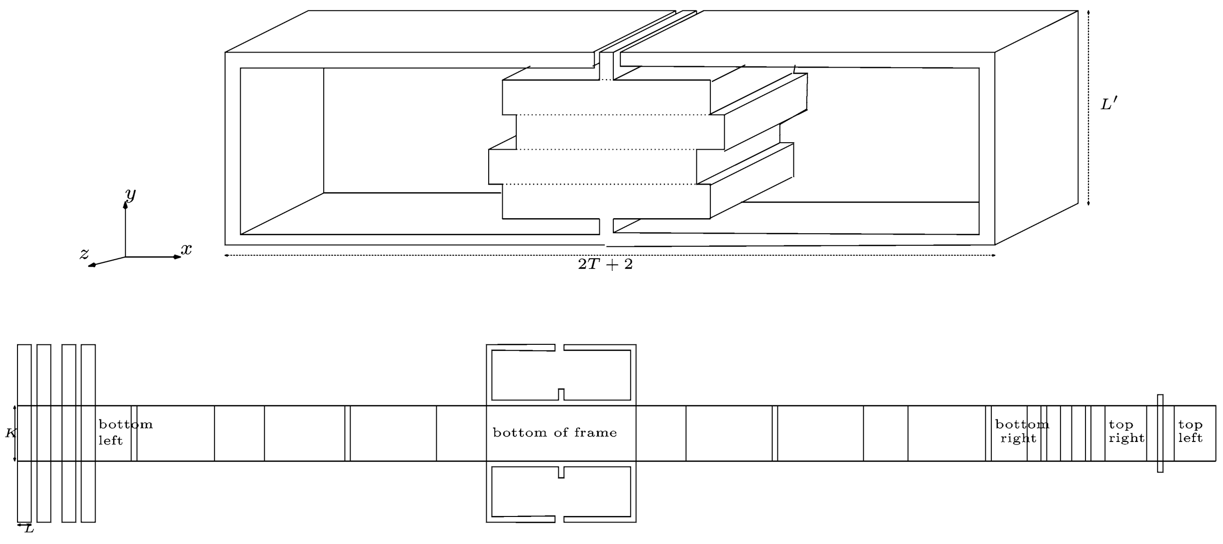

Fig. 5. Introducing middle notches and a frame.

Lemma 1. *If S has a solution, then S'' folds into a polyhedron. Specifically, S'' folds into an orthogonal polyhedron without dihedral angles of π.*

Now we want to show the converse.

Lemma 2. *If S'' folds into a polyhedron, then S has a balanced partition $S = S_1 \cup S_2$.*

Proof. Assume S'' folds into a polyhedron $\mathcal{P}$; we do not assume that $\mathcal{P}$ is orthogonal.

Let the rectangle labelled “bottom of frame” in Fig. 5 be B, and define the *spine* to be the rectangles in the horizontal row including B. Orient $\mathcal{P}$ so that B lies in the xz-plane with its top and bottom edges parallel to the x-axis at $z = 0$ and $z = K$ respectively. Regardless of the dihedral angles chosen for the creases of the spine (the ones that are vertical in Fig. 5), those edges will all be parallel to the z axis in $\mathcal{P}$. We claim that the exposed left and right edges of the spine in Fig. 5 must glue together. This is because none of the other faces can reach z-coordinate $K/2$: the back faces, which consist of $n+1$ strips, the back of the frame, and the back of the middle notch, all have an edge at $z = 0$, and have height at most L' which is less than $K/2$. The same argument applies to the front faces.

The spine therefore forms a “tube” or cycle of faces in $\mathcal{P}$. We now explore how the back and front faces fill in the open back and front ends of this tube, respectively. We concentrate on the back.

The back of the spine has 6 edges of length T or more. These edges cannot glue to each other. The total exposed perimeter of the back rectangles (i.e., excluding the frame) is $(n+1)(2L'+L)+2\frac{1}{2}$. This counts the lengths of the 3 exposed edges of each of the $n+1$ strips, and the 3 exposed edges of the middle notch. Because this total perimeter is less than T, thus all the exposed edges of the back rectangles are not sufficient to cover even one of the 6 long edges of the back of the spine. These long edges must therefore be glued to the corresponding long edges of the back of the frame. Thus the frame really does glue together as shown in Fig. 5.

We now claim that all the back strips lie in the plane $z = 0$. Any back strip has one short edge attached to the spine and lying in the plane $z = 0$ and thus the other short edge must also be parallel to the xy plane. Suppose the long edges of a back strip lie in direction d. If d is not parallel to the xy plane, then the long edges cannot glue to short edges of back strips, but only to other long edges of back strips. Because every back strip has a short edge in the $z = 0$ plane, this is only possible if d is parallel to the z axis. But then the short edge at the end of the strip can only glue to another short edge whose strip must then lie coincident with the first one, contradiction. Thus all back strips lie in the plane $z = 0$. Because the back of the frame is oriented with edges parallel to the x and y axes, thus all back rectangles are similarly oriented. This implies that the rectangles of the spine must also be oriented orthogonally.

The back rectangles must then lie inside the frame (the only exit has width $\frac{1}{2}$) and, since they are too tall to lie vertically (the inner height of the frame is less than L'), they must lie horizontally as in Figs. 2 or 3. Finally, the presence of the middle notch forces alignment, and therefore $\mathcal{P}$ provides a solution to the partition problem, as argued for the 2D case. □

The two lemmas together imply the main theorem.

Theorem 2. *The* 3*D folding problem is NP-hard, whether angles of* π *are allowed or not.*

We do not know whether this problem is NP-complete. Note that Section 3 relies on a real RAM for arbitrary dihedral angles, thus it does not imply a polynomial time verification step. However, the orthogonal version of the problem is NP-complete, because verification with basic arithmetic is then possible.

Theorem 3. *The 3D orthogonal folding problem is NP-complete, whether angles of* π *are allowed or not.*

5. Rigid nets

In this section, we show that if the net is made from stiff material, i.e., its faces are rigid, then we cannot always execute the folding process—from net to polyhedron—while keeping faces disjoint.

Theorem 4. *There exists a net and a polyhedron it can be folded to with the property that the folding cannot be performed while keeping faces rigid and their interiors disjoint.*

Proof. Consider the net shown on the left side in Fig. 6, which can be folded to the orthogonal polyhedron shown in Fig. 7. The ends of this "extruded chain" are supposed to be very long.

Imagine placing a chain on the faces that are shaded, as in the right picture of Fig. 6. Now, if we could fold this net of rigid faces without self-intersections, then we could also fold the chain without self-intersections into the position that it takes on the polyhedron (see the right picture of Fig. 7). Call this 3-dimensional chain K. It follows that K can be *straightened*, i.e., transformed without self-intersections into a straight chain: we first reverse the folding of the net to unfold K to the planar chain in Fig. 6, and then straighten it link by link. However, this leads to a contradiction because, using a proof very similar to the one in [7], we can show that if the end-links of K are sufficiently long, then it cannot be straightened,

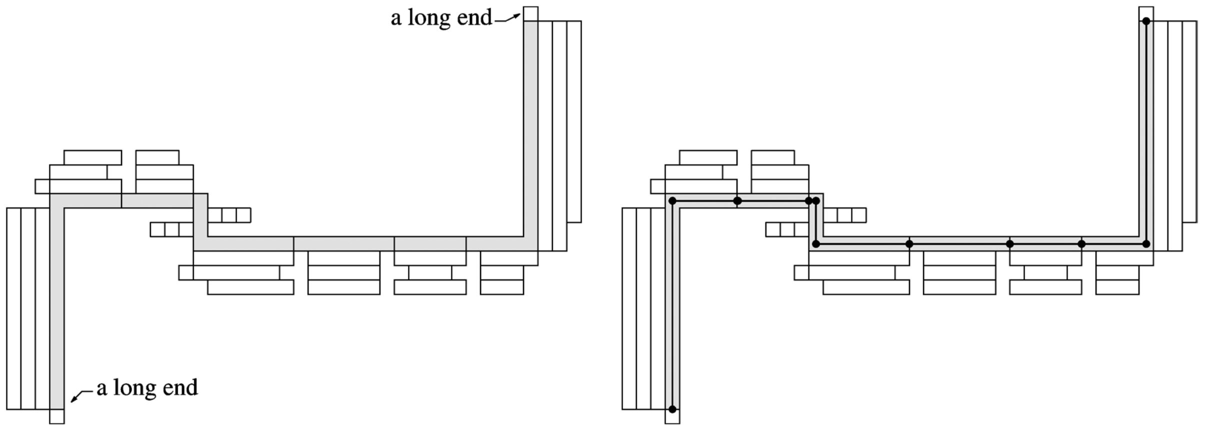

Fig. 6. The rigid net shown on the left cannot be folded without intersections.

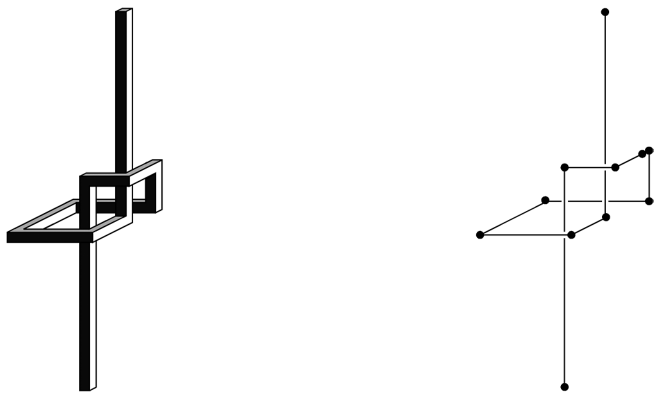

Fig. 7. The polyhedron for the rigid net, and the chain embedded on it.

even allowing arbitrary rotations of links, rather than just the one degree of freedom rotations possible for the chain on the net. □

6. Conclusions

In this paper, we studied the problem of determining whether a net folds into a polyhedron. We showed that if the dihedral angle at each crease is given, the question can be answered in polynomial time. However, if the dihedral angles are not given, the problem becomes NP-hard, and NP-complete for orthogonal polyhedra. Finally, we gave an example of a net (with dihedral angles given) that folds into a polyhedron, but where the folding cannot be performed if the net is made of stiff material (i.e., no extra creases are allowed) and faces may not intersect.

The fact that our NP-hardness result is based on an argument that a certain orthogonal polygon can fold only to an orthogonal polyhedron prompted us, in an earlier version of this paper [8], to pose the following question: Can a net where all faces are orthogonal ever fold to a non-orthogonal polyhedron?

More generally, is it true that if all faces of a polyhedron are orthogonal, then (after a suitable rotation), all faces of the polyhedron are perpendicular to a coordinate axis? We conjectured that the answer is yes. This has since been proved for polyhedra of genus $\leqslant 2$, and disproved for genus $\geqslant 6$. See [12] and [5].

We are also interested in other types of folding problems. For example, frequently with a net one is also given whether each crease is for a *mountain fold* or a *valley fold*, i.e., for a fold that bends away/towards the viewer. If this additional information is given, does the problem remain NP-hard? See [2] for related work on map folding.

Concerning rigid nets, we only gave one example. We do not know the complexity of the problem: given a net with rigid faces and dihedral angles for the creases, can the net be folded while keeping faces disjoint. It has recently been proved that the analogous problem for linkages in 3D is PSPACE-hard [1].

We mention one other problem concerning rigidity: can any feasible folding process for a net be accomplished with rigid non-intersecting faces if a finite number of extra creases (of dihedral angle π) may be introduced? See [15] for related results.

Acknowledgements

We thank an anonymous referee for finding an error in the NP-hardness reduction.

References

[1] H. Alt, C. Knauer, G. Rote, S. Whitesides, On the complexity of the linkage reconfiguration problem, in: Proceedings of the 19th Annual Symposium on Computational Geometry, San Diego, 2003, pp. 164–170.

[2] E.M. Arkin, M.A. Bender, E.D. Demaine, M.L. Demaine, J.S.B. Mitchell, S. Sethia, S.S. Skiena, When can you fold a map? Computational Geometry 29 (1) (2004) 23–46.

[3] B. Aronov, J. O'Rourke, Nonoverlap of the star unfolding, Discrete Comput. Geom. 8 (1992) 219–250.

[4] M. Bern, E. Demaine, D. Eppstein, E. Kuo, A. Mantler, J. Snoeyink, Ununfoldable polyhedra with convex faces, Comput. Geom. 24 (2) (2003) 51–62.

[5] T. Biedl, T. Chan, E. Demaine, M. Demaine, P. Nijjar, R. Uehara, M.-W. Wang, Tighter bounds on the genus of nonorthogonal polyhedra built from rectangles, in: Proceedings of the 14th Canadian Conference on Computational Geometry, 2002, pp. 105–108.

[6] T. Biedl, E. Demaine, M. Demaine, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, S. Whitesides, Unfolding some classes of orthogonal polyhedra, in: Proceedings of the 10th Canadian Conference on Computational Geometry, 1998, pp. 70–71.

[7] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, S. Whitesides, Locked and unlocked polygonal chains in 3D, Discrete Comput. Geom. 26 (3) (2001) 283–287.

[8] T. Biedl, A. Lubiw, J. Sun, When can a net fold to a polyhedron?, in: Proceedings of the 11th Canadian Conference on Computational Geometry, 1999, pp. 1–4.

[9] J. Cantarella, H. Johnston, Non-trivial embeddings of polygonal intervals and unknots in 3-space, J. Knot Theory Ramifications 7 (8) (1998) 1027–1039.

[10] H.S.M. Coxeter, Regular Polytopes, The Macmillan Company, 1963.

[11] P.R. Cromwell, Polyhedra, Cambridge Univ. Press, Cambridge, 1997.

[12] M. Donoso, J. O'Rourke, Nonorthogonal polyhedra built from rectangles, in: 14th Canadian Conference on Computational Geometry, 2002, pp. 97–100.

[13] A. Dürer, Unterweysung der Messung mit dem Zirckel uñ richtscheyt in Linien ebnen unnd ganzen corporen, Nürnberg, 1525, Translated and with a commentary by Walter L. Strauss in "The Painter's Manual", Abaris Books, New York, 1977.

[14] E. Demaine, D. Eppstein, J. Erickson, G. Hart, J. O'Rourke, Vertex-unfolding of simplicial manifolds, in: Proceedings of the 18th Annual ACM Symposium on Computational Geometry, 2002, pp. 237–243.
[15] E. Demaine, J.S.B. Mitchell, Reaching folded states of a rectangular piece of paper, in: Proceedings of the 13th Canadian Conference on Computational Geometry, 2001, pp. 73–75.
[16] H. Edelsbrunner, Algorithms in Combinatorial Geometry, Springer, Berlin, 1988.
[17] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, New York, 1979.
[18] A. Lubiw, J. O'Rourke, When can a polygon fold to a polytope?, Technical Report 048, Smith College, 1996.
[19] W.J. Lenhart, S.H. Whitesides, Reconfiguring closed polygonal chains in Euclidean d-space, Discrete Comput. Geom. 13 (1995) 123–140.
[20] G.C. Shephard, Convex polytopes with convex nets, Math. Proc. Camb. Philos. Soc. 78 (1975) 389–403.