Computational Complexity of Piano-Hinged Dissections

Zachary Abel† Martin L. Demaine‡
Erik D. Demaine† Takashi Horiyama§

Abstract

We prove NP-completeness of deciding whether a given loop of colored right isosceles triangles, hinged together at edges, can be folded into a specified rectangular three-color pattern. By contrast, the same problem becomes polynomially solvable with one color or when the target shape is a tree-shaped polyomino.

1 Introduction

One of the simplest and most practical physical folding structures is that of a hinge, as in most doors or attaching the lid to a grand piano. Frederickson [4] introduced a way to make folding structures out of such hinges that can change their shape between “nearly 2D” shapes. The basic idea is to thicken a (doubly covered) 2D polygon by extruding it orthogonally into a height-$2\epsilon$ 3D prism, divide that prism into two height-$\epsilon$ layers, further divide those layers into $\epsilon$-thickened polygonal pieces, and hinge the pieces together with hinges along shared edges. The goal in a piano-hinged dissection is to find a connected hinging of $\epsilon$-thickened polygonal pieces that can fold into two (or more) different $2\epsilon$-thickened polygons.

Piano-hinged dissections are meant to be a more practical form of hinged dissections, which typically use point hinges and thus are more difficult to build [4]. Although hinged dissections have recently been shown to exist for any finite set of polygons of equal area [1], no such result is known for piano-hinged dissections.

Here we study a family of simple piano-hinged dissections, which we call a piano-hinged loop: $4n$ identical $\epsilon$-thickened right isosceles triangles, alternating in orientation, and connected into a loop by hinges on the bottoms of their isosceles sides; see Figure 2. Frederickson [4, chapter 11] mentions without proof that this piano-hinged dissection can fold into any ($2\epsilon$-thickened) $n$-omino, that is, any connected edge-to-edge joining of $n$ unit squares.

Our results. In this paper, we investigate the computational complexity of folding colored and uncolored piano-hinged loop puzzles into $n$-ominos.

First we consider the uncolored piano-hinged loop, as in GeoLoop. For completeness, we prove Frederickson’s claim that this loop can realize any $2\epsilon$-thickened $n$-omino, by mimicking a simple inductive argument for hinged dissections of polyominoes from [2].

For the special case of tree-shaped polyominoes, where the dual graph of edge-to-edge adjacencies among unit squares forms a tree, we prove further that the folding of the piano-hinged loop is unique up to cyclic shifts of the pieces in the loop.

Next we consider colored piano-hinged loops, as in Ivan’s Hinge. For tree-shaped polyominoes, the previous uniqueness result implies that the problem can be solved in $O(n^2)$ time by trying all cyclic shifts. (In
2 Preliminaries

A piano-hinged loop consists of a loop of $4n$ consecutive isosceles right triangles $p_0, q_0, p_1, q_1, \ldots, p_{2n-1}, q_{2n-1}$, as shown in Figure 2. Every two consecutive triangular pieces share one of two isosceles edges. The $p_i$’s have a common orientation (collinear hypotenuses when unfolded), as do the $q_i$’s, and these two orientations differ from each other. Each shared edge is a piano hinge on the back side that permits bending inward (bringing the two back sides together).

In a folded state of the piano-hinged loop into a doubly covered polyomino, (1) each piano hinge is flat ($180^\circ$) or folded inward ($360^\circ$); and (2) each unit square of the polyomino consumes exactly four triangles, with two triangles on the front and two on the back side. Thus, in any folded state, the exposed surface consists of all front sides of the pieces, while the back sides of all pieces remain hidden on the inside. Therefore, we can ignore the color of the back side of each piece, so for simplicity we can assume that each piece has a uniform color (instead of a different color on each side). Let $c(p_i)$ and $c(q_i)$ denote the color of piece $p_i$ and $q_i$.

For the resultant polyomino $P$ of $n$ unit squares, we define the connection graph $G(P) = (V, E)$ as follows: $V$ consists of $n$ unit squares, and $E$ contains an edge $\{u, v\}$ if and only if squares $u$ and $v$ are adjacent (share an edge) in $P$. Having $\{u, v\} \in E$ is a necessary but not sufficient condition for there to be a hinge connecting the four pieces representing square $u$ to the four pieces representing square $v$; if there is such a hinge, we call $u$ and $v$ joined.

The uncolored piano-hinged loop problem asks whether a given polyomino can be constructed as (the silhouette of) a folded state of a given piano-hinged loop. The “silhouette” phrasing allows the folding to have unjoined squares, which are adjacent in the polyomino but not attached by a hinge in the folded state. The colored piano-hinged loop problem asks whether a given colored polyomino pattern can be similarly constructed from a given colored piano-hinged loop.

The piano-hinged loop has a simple checkerboarding property:

Observation 1 Consider two adjacent squares $u$ and $v$ in a polyomino $P$, obtained as a folded state of a piano-hinged loop. Without loss of generality, assume that the top side of $u$ contains (the front side of) triangle $p_i$. Then (1) the other triangle of $u$ on front side is $p_j$ for some $j$, (2) the backside of $u$ contains two $q$s, (3) the front side of $v$ contains two $p$s, and (4) the backside of $v$ contains two $q$s.

Ivan’s Hinge has a group of triangles that are monochromatic as assumed above, and a group of tri-
angles with different colors on their front and back sides. However, these groups directly correspond to the parity classes in Observation 1. Hence, for each unit square, the front side consists of two triangles from the same group, and the back side consists of two triangles from the other group. Thus, from a theoretical point of view, we can again effectively assume that the pieces are monochromatic. (Practically, the differing colors can vary the color patterns, which can help visually.)

3 Uncolored Piano-Hinged Loop

We begin with the universality theorem of GeoLoop, claimed by Frederickson [4]:

**Theorem 1** ([4]) Any polyomino $P$ of $n$ unit squares can be realized as a folded state of the piano-hinge loop of $4n$ pieces.

Once we fix the spanning tree $T$ of $G(P)$, we claim that the folded state is uniquely determined up to cyclic shift of the pieces. Both this corollary and the previous theorem follow from a simple argument of repeatedly pruning leaves in the graph of joinings.

**Corollary 2** Let $P$ be any polyomino of $n$ unit squares such that $G(P)$ is a tree. Then it can be uniquely folded from the piano-hinge loop of $4n$ pieces, up to cyclic shift of the pieces.

For a given tree-shaped polyomino, the piano-hinge loop traverses the tree in the same manner as a depth-first search without crossing. That is, if we imagine that we are in the maze in the form of the tree, and traverse the maze by the right-hand rule, then we traverse each edge twice, and this is the order followed by the piano-hinge loop. This intuition will be useful in some proofs in this paper.

4 General Piano-Hinged Loop

Consider a polyomino $P$ in which pieces $p_i$ and $q_i$ have colors $c(p_i)$ and $c(q_i)$, respectively. When the connection graph $G(P)$ is a tree (or the spanning tree of $G(P)$ is explicitly given), we still have a polynomial-time algorithm to solve the problem:

**Theorem 3** Let $P$ be any polyomino of $n$ unit squares such that $G(P)$ is a tree $T$. Then the general piano-hinge loop problem can be solved in $O(n^2)$ time.

Next we turn to the case that $P$ is a general polyomino, where the problem is NP-complete:

**Theorem 4** The colored piano-hinge loop problem is NP-complete, even if the number of colors is 3 and the target polyomino is a rectangle.
Note that this notion of “double covering” a rectangle allows cuts/seams in the middle of the rectangle (along grid edges). Without this flexibility, the folding would be uniquely determined up to cyclic shift, leading to an $O(n^2)$-time algorithm in the same manner as Theorem 3.

References


