Symmetric Assembly Puzzles are Hard, Beyond a Few Pieces

Erik D. Demaine[1], Matias Korman[2], Jason S. Ku[1], Joseph S. B. Mitchell[3], Yota Otachi[4], André van Renssen[5,6], Marcel Roeloffzen[5,6], Ryuhei Uehara[4], and Yushi Uno[7]

[1] MIT, {edemaine,jasonku}@mit.edu
[2] Tohoku University, mati@dais.is.tohoku.ac.jp.
[3] Stony Brook University, joseph.mitchell@stonybrook.edu
[4] JAIST, {otachi,uehara}@jaist.ac.jp
[5] National Institute of Informatics, {andre,marcel}@nii.ac.jp
[6] JST, ERATO, Kawarabayashi Large Graph Project
[7] Osaka Prefecture University, uno@mi.s.osakafu-u.ac.jp

Abstract. We study the complexity of symmetric assembly puzzles: given a collection of simple polygons, can we translate, rotate, and possibly flip them so that their interior-disjoint union is line symmetric? On the negative side, we show that the problem is strongly NP-complete even if the pieces are all polyominos. On the positive side, we show that the problem can be solved in polynomial time if the number of pieces is a fixed constant.

1 Introduction

The goal of a 2D *assembly puzzle* is to arrange a given set of pieces so that they do not overlap and form a target silhouette. The most famous example is the Tangram puzzle, shown in Fig. 1. Its earliest printed reference is from 1813 in China, but by whom or exactly when it was invented remains a mystery [5]. There are over 2,000 Tangram assembly puzzles [5], and many more similar 2D assembly puzzles [3]. A recent trend in the puzzle world is a relatively new type of 2D assembly puzzle which we call *symmetric assembly puzzles*. In these puzzles the target shape is not specified. Instead, the objective is to arrange the pieces so that they form a symmetric silhouette without overlap.

The first symmetric assembly puzzle, "Symmetrix", was designed in 2003 by Japanese puzzle designer Tadao Kitazawa and was distributed by Naoyuki Iwase as his exchange puzzle at the 2004 International Puzzle Party (IPP) in Tokyo [4]. In this paper, we aim for arrangements that are line symmetric (reflection through a line), but other symmetries such as rotational symmetry could also be considered. The lack of a specified target shape makes these puzzles quite difficult to solve.

We study the computational complexity of symmetric assembly puzzles in their general form. We define a *symmetric assembly puzzle* or SAP to be a set of k simple polygons $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$, with $n = |P_1| + \cdots + |P_k|$ the total

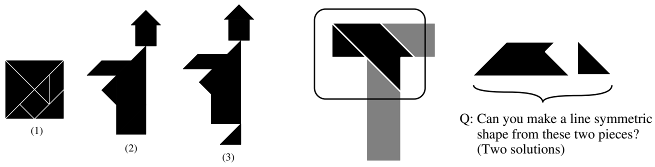

Fig. 1. [Left] The seven Tangram pieces (1) can be assembled into non-simple silhouettes (2) and (3). [Right] A symmetric assembly puzzle invented by Hiroshi Yamamoto [7]: given the two black pieces (right) from the classic T puzzle (left), make two different line symmetric shape. (Used with permission.)

number of vertices in all pieces. By *simple polygon* we mean a closed subset of $\mathbb{R}^2$ homeomorphic to a disk bounded by a closed path of straight line segments where nonadjacent edges and vertices do not intersect. A *symmetric assembly* $f : \mathcal{P} \to \mathbb{R}^2$ of a SAP $\mathcal{P}$ is a planar isometric embedding of the pieces so that their mapped interiors are disjoint and their mapped union forms a simple polygon that is line symmetric. We allow pieces to flip over (reflect), but other variants of the puzzle may disallow this. Given that humans have difficulty SAPs with even few low-complexity pieces, we consider two different generalizations: bounded piece complexity ($|P_i| = O(1)$) and bounded piece number ($k = O(1)$). In the former case, we prove strong NP-completeness, while in the latter case, we solve the problem in polynomial time (the exponent is linear in k).

2 Many Pieces

First we show that it is hard to solve symmetric assembly puzzles with a large number of pieces, even if each piece has bounded complexity ($|P_i| = O(1)$).

Theorem 1. *Symmetric assembly puzzles are strongly NP-complete even if each piece is a polyomino with at most six vertices and area upper bounded by a polynomial function of the number of pieces.*

If a SAP has a solution, the location and orientation of each piece within a symmetric assembly is a solution certificate of polynomial size checkable in polynomial time, so symmetric assembly puzzles are in NP. We reduce from the Rectangle Packing Puzzle problem, known to be strongly NP-hard [2]. Specifically, it is (strongly) NP-complete to decide whether k given rectangular pieces—sized $1\times x_1, 1\times x_2, \ldots, 1\times x_k$, where the x_i's are positive integers bounded above by a polynomial in k—can be exactly packed into a specified rectangular box with given width w and height h and area $x_1 + x_2 + \cdots + x_k = wh$.

Let $I = (x_1, \ldots, x_k, w, h)$ be a rectangle packing puzzle. Without loss of generality, we assume that $w \geq h$. Now let $I' = (P_1, \ldots, P_k, F)$ be the SAP where P_i is the $1 \times x_i$ rectangle for each $i \in \{1, \ldots, k\}$, and F is the polyomino in Fig. 2. We call F the *frame piece* of I'. We show that I has a rectangle packing if and only if I' has a symmetric assembly.

Clearly, if I has a rectangle packing, then the pieces $P_1, \ldots, P_k$ can be packed into the $w \times h$ hole in the frame piece creating a line symmetric $W \times H$ rectangle,

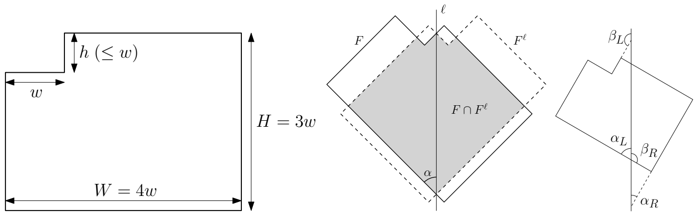

Fig. 2. [Left] The frame piece F. [Middle] If ℓ and ℓ_B form an angle of $\pi/4$, then $F \cap F^{\ell}$ is contained in a rectangle in an $H \times H$ and thus O^* cannot be line symmetric. [Right] The angles α_L, β_L, α_R, and β_R.

solving the SAP. Now we show the reverse implication. Assume that I' has a symmetric assembly, and let O^* be a line symmetric polygon formed by the pieces $\{P_1, \ldots, P_k, F\}$. We claim that O^* must be a $W \times H$ rectangle, which will imply that I is a yes-instance of RPP. Fix a placement of the pieces of I' that forms O^*, and let ℓ be one of its lines of symmetry. Assume, without loss of generality, that ℓ is a vertical line. Let F^{ℓ} be the reflection of F about ℓ.

Observation 1 $\mathsf{area}(F \cap F^{\ell}) \geq WH - 2wh \geq 10w^2$

Proof. Since O^* contains F^{ℓ} and F, it holds that $\mathsf{area}(F^{\ell} \setminus F) \leq \mathsf{area}(O^* \setminus F) = wh$. Since $F \cup F^{\ell}$ is mirror-symmetric, $\mathsf{area}(F^{\ell} \setminus F) = \mathsf{area}(F \setminus F^{\ell})$. Hence, it follows that $\mathsf{area}(F \cap F^{\ell}) = \mathsf{area}(F) - \mathsf{area}(F \setminus F^{\ell}) \geq WH - 2wh \geq 10w^2$. □

Observation 1 implies that ℓ passes through an interior point of F. Let ℓ_B be the line containing the segment of F with length $4w$. Let c be the center of the frame piece's bounding box.

Lemma 1. *ℓ_B is either parallel or orthogonal to ℓ.*

Proof. Suppose for contradiction that ℓ_B is neither parallel nor orthogonal to ℓ. Let α be the smaller angle made by ℓ_B and ℓ. We partition the edges of F crossed by ℓ into two at their intersection points. Let F_L and F_R be the sets of segments on the left and right portions of F, respectively. Consider the set of counter-clockwise angles between ℓ and the lines containing segments of F_L. The assumptions that ℓ_B and ℓ are neither parallel nor orthogonal, and that F is a polyomino together imply that the set contains exactly two angles α_L and β_L, where $\alpha_L \leq \beta_L$ and $\alpha_L + \pi/2 = \beta_L$. Similarly, let α_R and β_R be the clockwise angles between ℓ and the lines containing segments of F_R, where $\alpha_R \leq \beta_R$ and $\alpha_R + \pi/2 = \beta_R$. Since $\alpha_L + \beta_R = \pi$, it holds that $\alpha_L + \alpha_R = \pi/2$. Note that $\alpha \in \{\alpha_L, \alpha_R\}$.

Two distinct pieces of I' are *connected* if the fixed placement of the two pieces to form O^* have a non-degenerate line segment on their edges in common. Let $\mathcal{P}$ be the subset of $\{P_1, \ldots, P_n, F\}$ such that each $P_i \in \mathcal{P}$ can be reached from F by repeatedly following connected pieces in O^*.

As before, consider the angles formed by ℓ and the lines containing segments in the left and right parts of $\mathcal{P}$. Since all pieces are polyominoes, these lines cannot make angles other than α_L, β_L, α_R, and β_R with ℓ. Further note that the subset O' of O^* covered by $\mathcal{P}$ must be mirror-symmetric with respect to ℓ, or else O^* would not be. This implies that $\alpha_L = \alpha_R$. Since $\alpha_L + \alpha_R = \pi/2$, the only solution in which ℓ is not parallel or orthogonal to ℓ_B is when $\alpha_L = \alpha_R = \pi/4$ and $\alpha = \pi/4$. However, if $\alpha = \pi/4$, then $F \cap F^\ell$ is a subset of an $H \times H$ rectangle (see Fig. 2), whose area is at most $H^2 = 9w^2$, contradicting Observation 1. □

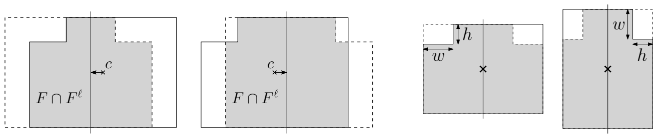

Fig. 3. [Left] When ℓ passes to the left of c, the portion of F to the left of ℓ is too small. If it passes to the right, the right portion would be too small. [Right] If ℓ passes through c, and is either orthogonal or parallel to ℓ_B, the symmetric assembly puzzle can only be completed into a rectangle.

So ℓ is either parallel or orthogonal to ℓ_B. Further, it passes through c (see Fig. 3). In either case, $F \cup F^\ell$ is a $W \times H$ rectangle, and thus $O^* = F \cup F^\ell$. This implies that $O^* \setminus F$ is a $w \times h$ rectangle that must contain the remaining pieces of I'. In particular, we have that this placement packing of $P_1, \ldots, P_n$ gives a solution to the instance I of RPP, completing the proof of Theorem 1.

3 Constant Pieces

Next we analyze symmetric assembly puzzles with a constant number of pieces but many vertices, and show they can be solved in polynomial time.

Theorem 2. *Given a symmetric assembly puzzle with a constant number of pieces k containing at most n vertices in total, deciding whether it has a symmetric assembly can be decided in polynomial time with respect to n.*

To prove this theorem, we present a brute force algorithm for solving a SAP that runs in polynomial time for constant k. We say two pieces in a symmetric assembly are *connected* to each other if their intersection in the symmetric assembly contains a non-degenerate line segment, and let the *connection* between two connected pieces be their intersection not including isolated points. We will call two pieces *fully* connected if their connection is exactly an edge of one of the pieces, and *partially* connected otherwise. Call a piece a *leaf* if it connects to at most one piece, and a *branch* otherwise. Given a leaf, let its *parent* be the piece connected to it (if it exists), and let its *siblings* be all other pieces connected to its parent. An illustration demonstrating these terms can be found in Fig. 4.

We will use a few utility functions in our algorithm. Deciding whether a single simple polygon has a line of symmetry can be done in linear time [6]. We will use `isSym(`P`)` to denote this algorithm, returning TRUE if polygon P has a line of symmetry and FALSE otherwise. In addition, we can test congruence of polygons in linear time using cong(P, Q), returning TRUE if P and Q are congruent polygons, and FALSE otherwise.

In addition, we will need to construct simple polygons from provided simple polygons by laying them next to each other along an edge. Let E_P denote the set of directed edges (p_i, p_j) from a vertex p_i to an adjacent vertex p_j of some simple polygon P. Given an edge $e \in E_P$, we denote its length by $\lambda(e)$. Let $e_P = (p_1, p_2)$ be a directed edge of a polygon P, let $e_Q = (q_1, q_2)$ be a directed edge of a polygon Q, and let d be a nonnegative length strictly less than $\lambda(e_P) + \lambda(e_P)$. Translate Q so that q_1 is incident to the point on the ray from p_1 containing e_P a distance d from p_1; then rotate Q so e_Q is collinear and in the same direction as e_P; and finally possibly reflect Q about e_Q if necessary so that the respective interiors of P and Q incident to e_P and e_Q lie in different half planes. Call these transformations the mapping $g : P \cup Q \to \mathbb{R}^2$. Then we define join$(e_P, e_Q, d)$ to be either, $g(P) \cup g(Q)$ if it is a simple polygon and the interior of $g(P) \cap g(Q)$ is empty (forms a simple polygon without overlapping pieces), or otherwise the empty set. See Fig. 4.

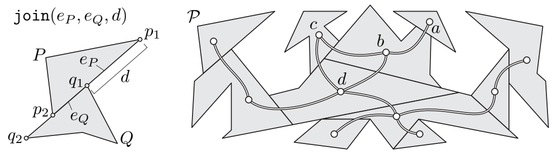

Fig. 4. [Left] Visualization of a join operation. [Right] Example symmetric assembly $\mathcal{P}$ showing its connection graph. Pieces a and d are fully connected to piece b, with c partially so. Pieces b, c, and d are branches. Piece a is a leaf, with b its parent and c and d the siblings of a.

If a SAP has a symmetric assembly, let its *connection graph* be a graph on the pieces with an edge connecting two pieces if they are connected in the symmetric assembly. Because a symmetric assembly is a simple polygon by definition, its connection graph is connected and has a spanning tree; we can then construct the assembly using a concatenation of join procedures in breadth-first-search order from an arbitrary root. Because parameter d is not discrete, the total solution space of simple polygons that are constructible from the pieces of a SAP may be uncountable. However, we can exploit the structure of symmetric assemblies to search only a finite set of configurations.

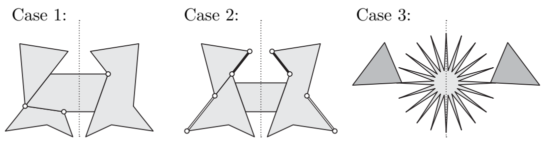

Fig. 5. Examples of symmetric assemblies belonging to each case. Case 1 highlights vertices of connected pieces that intersect. Case 2 highlights join operations using lengths of piece edges. Case 3 is constructed from one symmetric piece and a pair of congruent pieces.

In order to enumerate possible configurations, we would like to distinguish between three cases of puzzle (see Fig. 5), specifically:

Case 1: the puzzle has a symmetric assembly in which two connected pieces share a vertex on their connection;

Case 2: the puzzle has a symmetric assembly not satisfying Case 1 in which the distance between vertices from the connecting edges between two connected pieces has the same length as an edge from a third piece (we say the connection between two pieces *constructs* the length of another edge); or

Case 3: the puzzle has a symmetric assembly not satisfying Case 1 or Case 2 where a nonempty set of pieces are symmetric about the line of symmetry of the symmetric assembly, and any remaining pieces are pairs of congruent pieces.

Lemma 2. *If a SAP has a symmetric assembly, it can be described by one to the above three cases.*

Proof. Suppose for contradiction we have a symmetric assembly $f : \mathcal{P} \to \mathbb{R}^2$ of a SAP $\mathcal{P}$ that does not satisfy any of the above cases let $s : f(\mathcal{P}) \to f(\mathcal{P})$ be an automorphism reflecting $f(\mathcal{P})$ across a line of symmetry L, and let $\mu = s \circ f$, mapping a point $p \in \mathcal{P}$ to the reflection of $f(p)$ across L.

Consider the connection graph of $f(\mathcal{P})$. Because the symmetric assembly forms a simple polygon and no two connected pieces share a vertex, by exclusion from Case 1 the connection graph is a tree which we call a *connection tree*, or else the symmetric assembly would not be homeomorphic to a disk. Further, all connections are single non-degenerate line segments.

Let P be a leaf in the symmetric assembly, whose siblings include at most one branch. We claim that either P is a line symmetric polygon, or $\mu(P)$ is itself a piece of the SAP congruent to P contradicting exclusion from Case 3. First, if P has no parent and is the only piece in the symmetric assembly, P must be a line symmetric polygon. Otherwise, let Q be the parent of P with edge e_P from E_P touching edge e_Q from E_Q. Let e_{QP} denote the subset of e_Q that maps to the intersection $f(e_P) \cap f(e_Q)$. Segment $f(e_{QP})$ cannot lie along L or else one of $f(e_P)$ or $f(e_Q)$ would share a vertex with another piece, contradicting exclusion from Case 1. Alternatively suppose $f(e_{QP})$ and $\mu(e_{QP})$ are the same line segment. As a leaf, P connects to the rest of the symmetric assembly only through $f(e_{QP})$, so for the assembly to be symmetric, $f(P)$ must be the same as $\mu(P)$, and piece P is a line symmetric polygon.

Lastly, suppose $f(e_{QP})$ and $\mu(e_{QP})$ are not the same line segment; we claim $\mu(P)$ is itself a piece of the SAP congruent to P. Suppose for contradiction it were not. Then $\mu(P)$ either (a) contains a piece as a strict subset, (b) does not fully contain a piece but intersects interiors of multiple pieces, or (c) is a strict subset of a single piece (see Fig. 6).

First suppose (a), so $\mu(P)$ contains some piece S as a strict subset. Root the connection tree at a piece R with the shortest graph distance to S in the connection tree for which $f(R) \cap \mu(P) \neq \emptyset$ and $f(R) \setminus \mu(P) \neq \emptyset$ which exists

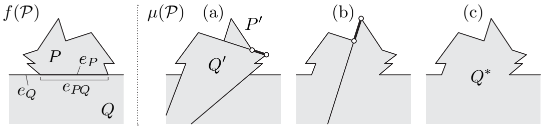

Fig. 6. Possible topological configurations of $\mu(P)$.

because $\mu(e_{PQ})$ must intersect some piece. Then a leaf P' with a longest root to leaf path that contains S is also fully contained in $\mu(P)$. Let Q' be its parent with edge e'_P from P' touching edge e'_Q from Q'. Because R is the piece crossing the boundary of $\mu(P)$ closest to S in the connection tree and P' has the longest root to leaf path, e'_Q connects to at most one branch piece that intersects $\mu(P)$. Segment $f(e'_P)$ cannot contain an edge of the symmetric assembly or else it would construct a length equal to an edge of P, contradicting exclusion from Case 2. So every leaf fully contained in $\mu(P)$ connected to e'_Q is fully connected to Q'. Each endpoint of the subset of e'_Q in $\mu(P)$ has shortest Euclidean distance to the connection of one leaf intersecting $\mu(P)$ connected to e'_Q. But at least one of these leaves is fully contained in $\mu(P)$ which that would construct a length equal to an edge of P, contradicting exclusion from Case 2. So $\mu(P)$ does not fully contain a leaf, contradicting case (a).

Now suppose (b), and suppose two connected pieces intersect $\mu(P)$. The edges connecting these two pieces must overlap in $\mu(P)$ to construct a length equal to an edge of P, contradicting exclusion from Case 2. So $\mu(P)$ does not intersect the interior of multiple branch pieces.

Finally suppose (c), and let $\mu(P)$ be the strict subset of some piece Q^*. Segment $f(e_P)$ cannot contain an edge of the symmetric assembly or else it would create a length equal to an edge of Q^*, contradicting exclusion from Case 2. So P is fully connected. A useful corollary of the preceding three arguments is that the reflection of any partially connected leaf of a symmetric assembly that conforms to neither Case 1 nor Case 2, must itself be a piece congruent to the leaf. We will refer to this property later as *partial leaf congruence.*

Here we note that none of the arguments so far have required P to be a leaf having at most one branch sibling; we will use that fact in the argument to follow. Let ℓ be the line collinear with segment $f(e_{QP})$, and let e_ℓ be the subset of Q that maps to the largest connected subset of $\ell \cap f(Q)$ containing $f(e_{QP})$. Consider the two disconnected sections of the boundary of Q between an endpoint of e_{PQ} and an endpoint of e_ℓ, which must each be more than an isolated point or exclusion from Case 1 would be violated. Piece P has at most one branch sibling, so at most one of these sections can be connected to a branch. Let q be an endpoint of e_ℓ in a section not connected to a branch.

Consider the boundary of Q between e_{QP} and q. Suppose this boundary were a line segment subset of e_Q, implying the internal angle of Q at q is less than π; see Fig. 7. Then $\mu(q)$ is in $f(Q^*)$ or else Q^* would connect to another piece somewhere on the segment between e_{QP} and q and construct an edge of the same length as a leaf connected to e_Q, contradicting exclusion from Case 2. If $\mu(q)$

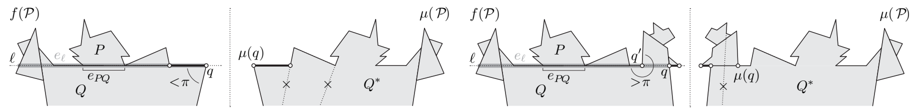

Fig. 7. Considering if $\mu(P)$ is a strict subset of Q^* and the boundary between e_{PQ} and q is a [Left] straight line or [Right] not a straight line.

is in $f(Q^*)$ and Q does not connect with any other piece at q, then $\mu(q)$ must be a vertex of $f(Q^*)$. Alternatively, q partially connects to a leaf through e_Q. By partial leaf congruence, the reflection of this leaf must itself be a congruent piece, so $\mu(q)$ is a vertex of $f(Q^*)$. In either case, the edge of Q^* adjacent to $\mu(q)$ contained in $\mu(e_Q)$ will have the same length as the subset of e_Q between q and a vertex of a leaf, contradicting exclusion from Case 2.

Thus, the boundary of Q between e_{QP} and q is not a line segment, so $f(Q)$ must cross ℓ, and the endpoint q' of e_Q in this section is a vertex of Q with internal angle greater than π; see Fig. 7. By the same argument as in the preceding paragraph, $\mu(q')$ must be in $f(Q^*)$, and if it were a vertex, we would have the same contradiction as before. However this time $\mu(q')$ need not be a vertex of $f(Q^*)$ because $f(Q^*)$ may extend past $\mu(q')$, with Q^* connecting to another piece on the other side of e_ℓ. However, the connection between these pieces will construct an edge that is the same length as an edge in either Q or a leaf connected to Q, and we have arrived at our final contradiction. So if P is not line symmetric, $\mu(P)$ is itself a piece of the SAP congruent to P.

Thus, our SAP has a leaf that is either a line symmetric piece, symmetric about the line of symmetry, and/or exists in a pair of two leaf pieces that are congruent and symmetric about the line of symmetry. If we remove such an identified leaf piece or pair from the SAP, what remains is a SAP with fewer pieces also admitting a symmetric assembly. Further, removing pieces cannot make the new SAP belong to one of the cases that the original SAP did not before. Repeatedly removing pieces using this process identifies every piece as either symmetric, or uniquely paired with a piece congruent to it, contradicting exclusion from Case 3. □

Since every symmetric assembly can be classified as one of these cases, we can check for each case to decide if the SAP has a symmetric assembly. Given a SAP that does not satisfy Case 1 or Case 2, by Lemma 2 it must satisfy Case 3 if it has a symmetric assembly. Satisfying Case 3 is not sufficient to ensure a symmetric assembly. For example, two congruent regular polygons with many sides and a single regular star with many spikes cannot by themselves form a symmetric assembly though they satisfy Case 3 because no pair of edges can be joined without making the pieces overlap. Thus given a SAP in Case 3, we must search the configuration space of possible connected arrangements of the pieces for an arrangement that forms a simple polygon.

Recall that the connection graph for a symmetric assembly not in Case 1 must be a tree. For a SAP with k pieces, Cayley's formula says the number of distinct connection trees is k^{k-2} [1]. However, even if two pieces are connected, they could be connected through $O(n^2)$ different pairs of edges, so the number of different *edge distinguishing connection trees*, connection trees distinguishing between which pairs of edges are connected, can be no more than $n^{2k}k^k = O(n^{2k})$ (k is constant). As an instance of Case 3, $\mathcal{P}$ consists of one or more symmetric pieces, with the rest being congruent pairs. Let $\mathcal{D}_\mathcal{P}$ and $\mathcal{D}'_\mathcal{P}$ be maximal disjoint subsets of $\mathcal{P}$ such that there exists a matching $\eta : \mathcal{D}'_\mathcal{P} \to \mathcal{D}_\mathcal{P}$ between pieces in $\mathcal{D}_\mathcal{P}$ and $\mathcal{D}'_\mathcal{P}$ such that matched pairs are congruent. Let $\mathcal{S}_\mathcal{P}$ be the set of symmetric pieces in $\mathcal{P}$ not in $\mathcal{D}_\mathcal{P}$ or $\mathcal{D}'_\mathcal{P}$. Let $\mathcal{S}_T$ denote some subset of the symmetric pieces contained in $\mathcal{D}_\mathcal{P}$, and define a *trunk* to be a subset of symmetric pieces $\mathcal{R}_T = \mathcal{S}_\mathcal{P} \cup \mathcal{S}_T \cup \eta(\mathcal{S}_T)$ that can be connected into a simple polygon without overlap while aligning each of their lines of symmetry to a common line L (see Fig. 8). Define a *half tree* T to be an edge distinguishing connection tree on $\mathcal{R}_T \cup \mathcal{D}_\mathcal{P}$ such that every piece in $\mathcal{D}_\mathcal{P}$ connected to a piece R in $\mathcal{R}_T$ connects through an edge of R intersecting the same half-plane bounded by L. We call this half-plane the *connecting half-plane*, with the other half-plane the *free half-plane*. The reason we define half trees is if we can find a point in their configuration space for which pieces do not intersect and for which pieces in $\mathcal{D}_\mathcal{P}$ not in the trunk do not intersect the free half-plane, we can place the remaining congruent pieces in $\mathcal{D}_\mathcal{P} \setminus \mathcal{S}_T$ at the mirror image of their respective matched pairs to complete a symmetric assembly.

Let $\mathcal{T}_\mathcal{P}$ be the set of possible half trees. Let $\mathcal{L}_T$ be the set of undirected edges $\{P, Q\}$ where piece P is connected to piece Q in tree $T \in \mathcal{T}_\mathcal{P}$, and let $m = |\mathcal{L}_T| < k$. For a fixed edge distinguishing connection tree, the orientation of each piece is fixed as pieces may only translate along their specified connection. We want to define a set of intervals $\mathcal{I}_T\{P, Q\}$ where we could join e_P to e_Q while together forming a simple polygon, without overlap between P and Q. For each $\{P, Q\} \in \mathcal{L}_T$ with e_P and e_Q the respective connecting edges of P and Q with $\lambda(e_P) \geq \lambda(e_Q)$, let $\mathcal{I}_T\{P, Q\}$ be defined as follows. If P and Q are both in $\mathcal{R}_T$, let $\mathcal{I}_T\{P, Q\}$ be the empty set if $\texttt{join}(e_P, e_Q, d_{PQ})$ is the empty set and $\{d_{PQ}\}$ otherwise, where we use d_{PQ} to denote $|\lambda(e_P) - \lambda(e_Q)|/2$, the distance d would need to be in order to align the midpoints of e_P and e_Q. Alternatively if P or Q are not in $\mathcal{R}_T$, let $\mathcal{I}_T\{P, Q\}$ be the closure of the set of distances d for which $\texttt{join}(e_P, e_Q, d)$ is nonempty. The number of distinct intervals in $\mathcal{I}_T\{P, Q\}$ is at most linear in the number of vertices, $O(n)$. Any fixed arrangement of the pieces consistent with edge distinguishing connection tree T joins each pair of pieces by fixing one point in every $\mathcal{I}_T\{P, Q\}$, so the set of configurations is a subset of $\mathbb{R}^m$. Ignoring overlap between pieces that are not connected, the configuration space $\mathcal{C}_T$ of possible arrangements is equal to the cartesian product of $\mathcal{I}_T\{P, Q\}$ for every $\{P, Q\} \in \mathcal{L}_T$. Thus $\mathcal{C}_T$ is a set of $O(n^m)$ disjoint m-dimensional hyperrectangles in $\mathbb{R}^m$.

We now describe the subset of $\mathbb{R}^m$ where intersection occurs between two pieces that are not connected in T. If two pieces in a configuration overlap, by continuity there exist two edges e_P and e_Q from two distinct pieces P and Q that also intersect. The positions of e_P and e_Q are translations parameterized by a point in $\mathcal{C}_T$ and the region in which the two edges intersect is a convex region

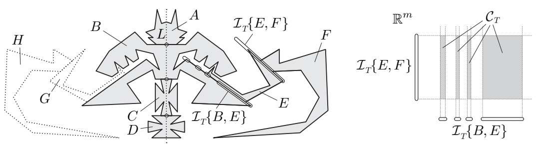

Fig. 8. An example showing a SAP $\mathcal{P}$ satisfying Case 3, with $\mathcal{S}_\mathcal{P} = \{A, B\}$, $\mathcal{D}_\mathcal{P} = \{C, E, F\}$, $\mathcal{D}'_\mathcal{P} = \{D, G, H\}$, $\mathcal{S}_T = \{C\}$, $\eta(\mathcal{S}_T) = \{D\}$, and trunk $\mathcal{R}_T = \{A, B, C, D\}$. $\mathcal{I}_T$ for two connected pieces in the trunk is just a single point as shown by the midpoint of their connection. Pieces not in the trunk have a degree of freedom sliding along their connection. $\mathcal{I}_T\{E, F\}$ is a single interval where F can attach to E, while $\mathcal{I}_T\{B, E\}$ is a four intervals. The right diagram shows $\mathcal{C}_T$ the cartesian product of each $\mathcal{I}_T$.

Function `hasAssemblyCase3(`$\mathcal{P}$`)`
  input : Symmetric assembly puzzle $\mathcal{P}$ that satisfies Case 3.
  output : TRUE if $\mathcal{P}$ has a symmetric assembly, FALSE otherwise.
  for $T \in \mathcal{T}_\mathcal{P}$ **do**
    $\mathcal{C}'_T \leftarrow \mathcal{C}_T$
    for $\{P, Q\} \in \mathcal{L}_T$ **do**
      $\mathcal{C}'_T \leftarrow \mathcal{C}'_T \setminus \mathcal{X}_T\{e_P, e_Q\}$
    if $\text{int}(\mathcal{C}'_T) \neq \emptyset$ **then**
      return TRUE
    else if $\mathcal{C}'_T \neq \emptyset$ **and** $\mathcal{D}_\mathcal{P} = \emptyset$ **then**
      return TRUE
  return FALSE

Algorithm 1: Pseudocode for function `hasAssemblyCase3(`$\mathcal{P}$`)`

$\mathcal{X}_T\{e_P, e_Q\} \subset \mathbb{R}^m$ bounded by four hyperplanes forming the m-dimensional parallelogram representing the intersection of the two edges. For each $O(n^2)$ pair of edges from distinct pieces that are not connected, we can subtract each $\mathcal{X}_T\{e_P, e_Q\}$ from $\mathcal{C}_T$ to form $\mathcal{C}'_T$. If $\mathcal{C}'_T$ contains any point in its interior, then there exists a symmetric assembly since it will be a point in the configuration space avoiding overlap between pieces. However, the boundary of $\mathcal{C}'_T$ may contain configurations that are weakly simple as the boundaries of each $\mathcal{I}_T$ not between two pieces in $\mathcal{R}_T$ and the boundaries of each $\mathcal{X}_T$ all correspond to configuration containing non-simple touching between pieces. Thus we require $\mathcal{C}'_T$ to have a point on its interior unless all pieces exist in R_T, where $\mathcal{C}'_T$ may be a single point corresponding to a symmetric assembly.

Consider the function `hasAssemblyCase3` described in Algorithm 1.

Lemma 3. *Given symmetric assembly puzzle $\mathcal{P}$ that satisfies Case 3, function* `hasAssemblyCase3(`$\mathcal{P}$`)` *returns* TRUE *if and only if $\mathcal{P}$ has a symmetric assembly, and terminates in $O(n^{5k})$ time.*

Proof. If $\mathcal{P}$ has a symmetric assembly satisfying Case 3 with nonempty $\mathcal{D}_\mathcal{P}$, $\mathcal{C}'_T$ will have a point on its interior for some tree T as argued above; or if $\mathcal{D}_\mathcal{P}$ is empty, $\mathcal{C}'_T$ will be nonempty. There are $O(n^{2k})$ elements of $\mathcal{T}_\mathcal{P}$. There are $m = O(k)$ interval sets $\mathcal{I}_T\{P, Q\}$ each having computational complexity $O(n)$, so we can construct $\mathcal{C}_T$ naively in $O(n^k)$ time. The union X_T of the $O(n^2)$ regions $\mathcal{X}_T\{e_P, e_Q\}$, which are m-dimensional convex regions, has computational complexity at most $O(n^{2m})$, so the final computational complexity of $\mathcal{C}'_T = \mathcal{C}_T \setminus X_T$ is at most $O(n^{3m})$ and can be computed in as much time. Thus, the running time of `hasAssemblyCase3` is bounded by $O(n^{5k})$. □

```
Function hasAssembly(𝒫)
  input  : Symmetric assembly puzzle 𝒫.
  output : TRUE if 𝒫 satisfies Case 1 or Case 2 or Case 3, FALSE otherwise.
  for e_P ∈ E_P, e_Q ∈ E_Q, {P, Q} ⊂ 𝒫 do
    S ← join(e_P, e_Q, 0)
    𝒫' ← (𝒫 \ {P, Q}) ∪ {S}
    if S ≠ ∅ and hasAssembly(𝒫') then
      return TRUE                                         // Case 1
    for e_R ∈ E_R, R ∈ 𝒫 do
      if λ(e_R) < λ(e_P) then
        S ← join(e_P, e_Q, λ(e_R))
        𝒫' ← (𝒫 \ {P, Q}) ∪ {S}
        if S ≠ ∅ and hasAssembly(𝒫') then
          return TRUE                                     // Case 2
  return hasAssemblyCase3(𝒫)                              // Case 3
```

Algorithm 2: Pseudocode for function `hasAssembly`($\mathcal{P}$)

Our brute force algorithm `hasAssembly(`$\mathcal{P}$`)` is described in Algorithm 2.

Lemma 4. *Function* `hasAssembly(`$\mathcal{P}$`)` *returns* TRUE *if and only if* $\mathcal{P}$ *has a symmetric assembly that satisfies either Case 1, Case 2, or Case 3, and terminates in* $O(n^{5k})$ *time.*

Proof. We prove by induction. For the base case, $\mathcal{P}$ consists of only a single piece satisfying Case 3, which will drop directly to the last line of the algorithm checking Case 3 which, by Lemma 3 will evaluate correctly. Now suppose `hasAssembly` returns a correct evaluation for SAPs containing $k-1$ pieces. Then we show `hasAssembly` returns a correct evaluation for SAPs containing k pieces.

The outer **for** loop of `hasAssembly` cycles through every pair of directed edges $e_P = (p_1, p_2)$ and $e_Q = (q_1, q_2)$ taken from different pieces P and Q. For each pair, `hasAssembly` first checks to see if there exists a symmetric assembly for which e_P is connected to e_Q with p_1 coincident to q_1, which would satisfy Case 1. If one exists, then joining P and Q into one piece as described would produce a SAP $\mathcal{P}'$ with one fewer piece that also has a symmetric assembly. Then evaluating `hasAssembly` on the smaller instance will return correctly by induction. Since the outer **for** loop checks every possible pair of edges that could be joined in a symmetric assembly satisfying Case 1, `hasAssembly` will return TRUE if $\mathcal{P}$ satisfies Case 1.

Next `hasAssembly` checks to see if there exists a symmetric assembly for which e_P is connected to e_Q with p_1 and q_1 separated by a distance equal to the length of some other edge e_R in $\mathcal{P}$, which would satisfy Case 2. In the same way as with Case 1, both **for** loops check every possible pair of edges and that could be joined at every possible length that could produce a symmetric assembly satisfying Case 2, so `hasAssembly` will return TRUE if $\mathcal{P}$ satisfies Case 2.

Otherwise, no symmetric assembly exists satisfying Case 1 or Case 2. By Lemma 3, `hasAssemblyCase3` correctly evaluates if $\mathcal{P}$ is in Case 3, so `hasAssembly` returns a correct evaluation for SAPs containing k pieces. Let $T(k)$ be the running time of `hasAssembly` on an instance with k pieces. Then the recurrence relation for `hasAssembly` is $T(k) = O(n^3)T(k-1) + O(n^{5k})$, where $O(n^{5k})$ is the running time given by Lemma 3. Running time for Case 3 dominates the recurrence relation so `hasAssembly` terminates in $O(n^{5k})$. □

Now we can determining whether a symmetric assembly puzzle with a constant number of pieces has a symmetric assembly in polynomial time.

Proof (of Theorem 2). By Lemma 2, if the SAP has a symmetric assembly, it satisfies either Case 1, Case 2, or Case 3, and by Lemma 4 `hasAssembly(`$\mathcal{P}$`)` can correctly determine if it has a symmetric assembly satisfying one of the cases in polynomial time, proving the claim. □

Open questions include whether SAPs: are hard for simpler shapes (we conjecture SAPs containing only right triangles are still hard), are hard for non-simple target shapes with constant pieces, or are fixed-parameter tractable with respect to the number of pieces (we conjecture W[1]-hardness).

Acknowledgements: Many of the authors were introduced to symmetric assembly puzzles during the 30th Winter Workshop on Computational Geometry at the Bellairs Research Institute of McGill University, March 2015. Korman is supported in part by the ELC project (MEXT KAKENHI No. 24106008). Mitchell is supported in part by the National Science Foundation (CCF-1526406). Uno is supported in part by the ELC project (MEXT KAKENHI No. 15H00853).

References

[1] A. Cayley. A theorem on trees. *Quart. J. Math*, 23:376–378, 1889.

[2] E. D. Demaine and M. L. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 23(Supplement):195–208, 2007.

[3] E. Fox-Epstein and R. Uehara. The convex configurations of “Sei Shonagon Chie no Ita” and other dissection puzzles. In *26th Canadian Conference on Computational Geometry (CCCG 2014)*, pages 386–389, 2014.

[4] N. Iwase. Symmetrix. In *24th International Puzzle Party (IPP 24)*, page 54. IPP24 Committee, unpublished, 2005.

[5] J. Slocum. *The Tangram Book: The Story of the Chinese Puzzle with Over 2000 Puzzle to Solve*. Sterling Publishing, 2004.

[6] J. D. Wolter, T. C. Woo, and R. A. Volz. Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer*, 1(1):37–48, 1985.

[7] H. Yamamoto. Personal communication. 2014.