

# New Geometric Algorithms for Fully Connected Staged Self-Assembly

Erik D. Demaine<sup>1</sup>, Sándor P. Fekete<sup>2</sup>(✉),  
Christian Scheffer<sup>2</sup>, and Arne Schmidt<sup>2</sup>

<sup>1</sup> CSAIL, MIT, Cambridge, MA, USA  
edemaine@mit.edu

<sup>2</sup> Department of Computer Science, TU Braunschweig, Braunschweig, Germany  
{s.fekete,c.scheffer,arne.schmidt}@tu-bs.de

**Abstract.** We consider *staged self-assembly systems*, in which square-shaped tiles can be added to bins in several stages. Within these bins, the tiles may connect to each other, depending on the *glue types* of their edges. Previous work by Demaine et al. showed that a relatively small number of tile types suffices to produce arbitrary shapes in this model. However, these constructions were only based on a spanning tree of the geometric shape, so they did not produce full connectivity of the underlying grid graph in the case of shapes with holes; designing fully connected assemblies with a polylogarithmic number of stages was left as a major open problem. We resolve this challenge by presenting new systems for staged assembly that produce fully connected polyominoes in  $\mathcal{O}(\log^2 n)$  stages, for various scale factors and temperature  $\tau = 2$  as well as  $\tau = 1$ . Our constructions work even for shapes with holes and uses only a constant number of glues and tiles. Moreover, the underlying approach is more geometric in nature, implying that it promised to be more feasible for shapes with compact geometric description.

## 1 Introduction

In *self-assembly*, a set of simple *tiles* form complex structures without any active or deliberate handling of individual components. Instead, the overall construction is governed by a simple set of rules, which describe how mixing the tiles leads to bonding between them and eventually a geometric shape.

The classic theoretical model for self-assembly is the *abstract tile-assembly model* (aTAM). It was first introduced by Winfree [12, 14]. The *tiles* used in this model are building blocks, which are unrotatable squares with a specific glue on each side. Equal glues have a connection strength and may stick together. The *glue complexity* is the number of different glues, while the *tile complexity* is the number of tile types. If an additional tile wants to attach to the existing assembly by making use of matching glues, the sum of corresponding glue strengths needs to be at least some minimum value  $\tau$ , which is called the *temperature*.

A generalization of the aTAM called the *two-handed assembly model* (2HAM) was introduced by Demaine et al. [4]. While in the aTAM, only individual tiles

can be attached to an existing intermediate assembly, the 2HAM allows attaching other partial assemblies. If two partial assemblies (“supertiles”) want to assemble, then the sum of the glue strength along the whole common boundary needs to be at least  $\tau$ .

In this paper we consider the *staged tile assembly model* introduced in [4], which is based on the 2HAM. In this model the assembly process is split into sequential stages that are kept in separate bins, with supertiles from earlier stages mixed together consecutively to gain some new supertiles. We can either add a new tile to an existing bin, or we pour one bin into another bin, such that the content of both get mixed. Hence, there are bins at each stage. Unassembled parts get removed. The overall number of necessary stages and bins are the *stage complexity* and the *bin complexity*. Demaine et al. [4] achieved several results summarized in Table 1. Most notably, they presented a system (based on a spanning tree) that can produce arbitrary polyomino shapes  $P$  in  $\mathcal{O}(\text{diameter})$  many stages,  $\mathcal{O}(\log N) = \mathcal{O}(\log n)$  bins and a constant number of glues, where  $N$  is the number of tiles of  $P$ ,  $n$  is the size of a smallest square containing  $P$ , and the diameter is measured by a shortest path within  $P$ , so it can be as big as  $N$ . The downside is that the resulting shapes are not fully connected. For achieving full connectivity, only the special case of monotone shapes was resolved by a system with  $\mathcal{O}(\log n)$  stages; for hole-free shapes, they were able to give a system with full connectivity, scale factor 2, but  $\mathcal{O}(n)$  stages. This left a major open problem: designing a staged assembly system with full connectivity, polylogarithmic stage complexity and constant scale factor for general shapes.

**Our Results.** We show that for any polyomino, even with holes, there is a staged assembly system with the following properties, both for  $\tau = 2$  and  $\tau = 1$ .

1. polylogarithmic stage complexity,
2. constant glue and tile complexity,
3. constant scale factor,
4. full connectivity.

See Table 1 for an overview. The main novelty of our method is to focus on the underlying geometry of a constructed shape  $P$ , instead of just its connectivity graph. This results in bin numbers that are a function of  $k$ , the number of vertices of  $P$ : while  $k$  can be as big as  $\Theta(n^2)$ ,  $n$  can be arbitrarily large for fixed  $k$ , implying that our approach promises to be more suitable for constructing natural shapes with a clear geometric structure.

**Related Work.** As mentioned above, our work is based on the 2HAM. There is a variety of other models, e.g., see [2]. A variation of the staged 2HAM is the *Staged Replication Assembly Model* by Abel et al. [1], which aims at reproducing supertiles by using *enzyme self assembly*. Another variant is the *Signal Tile Assembly Model* introduced by Padilla et al. [9].

Other related geometric work by Cannon et al. [3] and Demaine et al. [5] considers reductions between different systems, often based on geometric properties. Fu et al. [7] use geometric tiles in a generalized tile assembly model to

**Table 1.** Overview of results from [4] and this paper. The number of tiles of  $P$  is denoted by  $N \in \mathcal{O}(n^2)$ ,  $n$  is the side length of a smallest bounding square, while  $k$  is the number of vertices of the polyomino, with  $k \in \Omega(1)$  and  $k \in \mathcal{O}(N)$ .

Lines and Squares	Glues	Tiles	Bins	Stages	$\tau$	Scale	Conn.	Planar
Line [4]	3	6	7	$\mathcal{O}(\log n)$	1	1	full	yes
Square — Jigsaw techn. [4]	9	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\log n)$	1	1	full	yes
Square — $\tau = 2$ ( <b>Sect. 2.1</b> )	4	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\log n)$	2	1	full	yes

Arbitrary Shapes	Glues	Tiles	Bins	Stages	$\tau$	Scale	Conn.	Planar
Spanning Tree Method [4]	2	16	$\mathcal{O}(\log n)$	$\mathcal{O}(\text{diameter})$	1	1	partial	no
Monotone Shapes [4]	9	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$	1	1	full	yes
Hole-Free Shapes [4]	8	$\mathcal{O}(1)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	1	2	full	no
Shape with holes ( <b>Sect. 2.2</b> )	7	$\mathcal{O}(1)$	$\mathcal{O}(k)$	$\mathcal{O}(\log^2 n)$	2	3	full	no
Hole-Free Shapes ( <b>Sect. 2.2</b> )	7	$\mathcal{O}(1)$	$\mathcal{O}(k)$	$\mathcal{O}(\log n)$	2	3	full	no
Hole-Free Shapes ( <b>Sect. 3.1</b> )	18	$\mathcal{O}(1)$	$\mathcal{O}(k)$	$\mathcal{O}(\log^2 n)$	1	4	full	no
Shape with holes ( <b>Sect. 3.2</b> )	20	$\mathcal{O}(1)$	$\mathcal{O}(k)$	$\mathcal{O}(\log^2 n)$	1	6	full	no

assemble shapes. Fekete et al. [6] study the power of using more complicated polyominoes as tiles.

Using stages has also received attention in DNA self assembly. Reif [11] uses a stepwise model for parallel computing. Park et al. [10] consider assembly techniques with hierarchies to assemble DNA lattices. Somei et al. [13] use a stepwise assembly of DNA tiles. Padilla et al. [8] include active signaling and glue activation in the aTAM to control hierarchical assembly of Robinson patterns. None of these works considers complexity aspects.

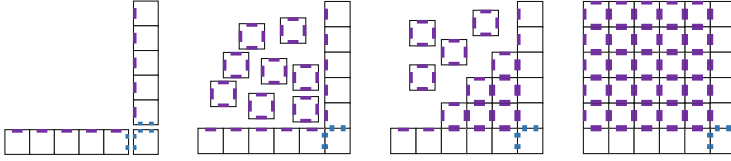
## 2 Fully-Connected Constructions for $\tau = 2$

In the following, we consider fully connected assemblies for temperature  $\tau = 2$ . We start by an approach for squares (Sect. 2.1). In Sect. 2.2 we describe how to extend this basic idea to assembling general polyominoes.

### 2.1 $n \times n$ Squares, $\tau = 2$

For  $\tau = 2$  assembly systems, it is possible to develop more efficient ways for constructing a square. The construction is based on an idea by Rothmund and Winfree [12], which we adapt to staged assembly. Basically, it consists of connecting two strips by a corner tile, before filling up this frame; see Fig. 1.

**Theorem 1.** *There exists a  $\tau = 2$  assembly system for a fully connected  $n \times n$  square with  $\mathcal{O}(\log n)$  stages, 4 glues, 14 tiles and 7 bins.*



**Fig. 1.** Construction of fully connected square using  $\tau = 2$  and a frame.

*Proof.* The construction is an easy result of combining known construction for lines by staged assembly with filling in squares in the aTAM with temperature  $\tau = 2$ , as follows: First we construct the  $1 \times (n - 1)$  strips with strength-2 glues. We know from [4] that a strip can be constructed in  $\mathcal{O}(\log n)$  stages, three glues, six tiles and seven bins. Because both strips are perpendicular, they will not connect. Therefore, we can use all seven bins to construct both strips in parallel. For each strip we use tiles such that the edge toward the interior of the square has a strength-1 glue. In the next stage we mix the single corner tile with the two strips. Finally, we add a tile type with strength-1 glues on all sides. When the square is filled, no further tile can still connect, as  $\tau = 2$ .

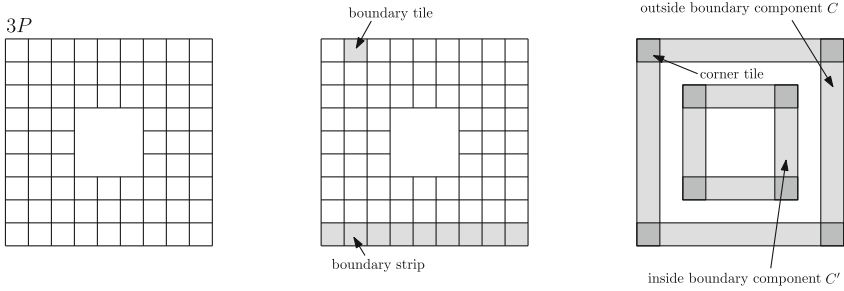
Overall, we need  $\mathcal{O}(\log n)$  stages with four glues (three for the construction, one for filling up the square), 14 tiles (six for each of the two strips, one for the corner tile, one for filling up the square) and seven bins for the parallel construction of the two strips.  $\square$

## 2.2 Polygons with or Without Holes, $\tau = 2$

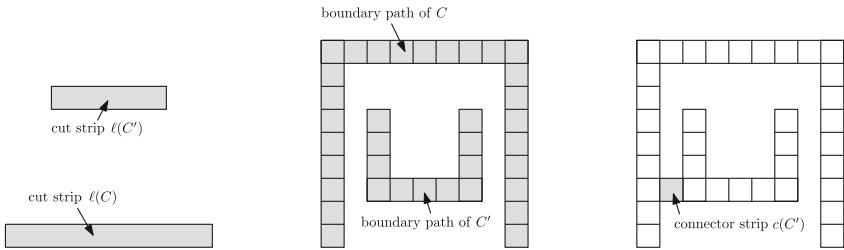
Our method for assembling a polyomino  $P$  at  $\tau = 2$  generalizes the approach for building a square that is described in Sect. 2.1. The key idea is to scale  $P$  by a factor of 3, yielding  $3P$ ; for this we first build a frame called the *backbone*, which is a spanning tree based on the union of all boundaries of  $3P$ . This backbone is then filled up in a final stage by applying a more complex version of the flooding approach of Theorem 1. In particular, there is not only one flooding tile, but a constant set  $S$  of such distinct tiles.

**Definition and Construction of the Backbone.** In the following, we consider a scaled copy  $3P$  of a polyomino  $P$ , constructed by replacing each tile by a  $3 \times 3$  square of tiles. We define the *backbone* of  $3P$  as follows; see Fig. 2 for an illustration.

**Definition 1.** A *tile* of  $3P$  is a boundary tile of  $3P$ , if one of the tiles in its eight (axis-parallel or diagonal) neighbor tiles does not belong to  $3P$ . A *boundary strip* of  $3P$  is a maximal set of boundary tiles that forms a contiguous (vertical or horizontal) strip, see Fig. 2(b). A *boundary component*  $C$  is a connected component of boundary tiles; because of the scaling, an inside boundary component corresponds to precisely one inside boundary of  $3P$  (delimiting a hole), while the outside boundary component corresponds to the exterior boundary of  $3P$ , see Fig. 2(c). Furthermore, each boundary component  $C$  has a unique decomposition



(a) 3-scaled polyomino  $3P$ . (b) Boundary tiles and strips. (c) Outside and inside boundary components.



(d) Cut strip of  $C$  and  $C'$ . (e) Boundary paths. (f) Connector strip  $c(C')$ .

**Fig. 2.** Stepwise construction of the backbone of a 3-scaled polyomino  $3P$ .

into boundary strips: a circular sequence of boundary strips that alternate between vertical and horizontal, with consecutive strips sharing a single (“corner”) tile. For an inside boundary component  $C'$ , its cut strip  $\ell(C')$  is the leftmost of its topmost strips; for the outside boundary component, its cut strip  $\ell(C)$  is the leftmost of its bottommost strips, see Fig. 2(d).

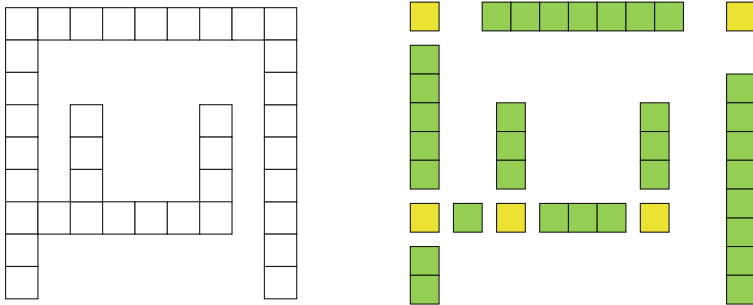
A boundary path of the outside boundary component  $C$  consists of the union of all its strips, with the exception of  $\ell(C)$ ; for an inside boundary component  $C'$ , it consists of  $C' \setminus \ell(C')$ ; see Fig. 2(e). Furthermore, the connector strip  $c(C')$  for an inside boundary component  $C'$  is the contiguous horizontal set of tiles of  $3P$  extending to the left from the leftmost bottommost tile of  $C'$  and ending with the first encountered other boundary tile of  $3P$ ; see Fig. 2(f). Then the backbone of  $3P$  is the union of all boundary paths and the connector strips of inside boundary components.

By construction, the backbone has a canonical decomposition into boundary strips and connector strips; furthermore, a tile in the backbone of  $3P$  is part of three different strips if and only if it is an end tile of a connector strip. For  $h$  holes, only  $2h$  tiles in the backbone are part of three different strips.

Overall, this yields a hole-free shape that can be constructed efficiently.

**Lemma 1.** *Let  $k$  be the number of vertices of a 3-scaled polyomino  $3P$ . The corresponding backbone can be assembled in  $\mathcal{O}(\log^2 n)$  stages with 4 glues,  $\mathcal{O}(1)$  tiles and  $\mathcal{O}(k)$  bins.*

*Proof.* The main idea is to give a recursive separation of the backbone into trivial tiles such that its reversed order implies a staged self-assembly that fulfills the required guarantees. For separating the backbone, we observe that it consists of two types of components: strips and corner tiles (see Fig. 3). The degree of such a corner tile is two or three, corresponding to the number of adjacent strips.



**Fig. 3.** (Left) A backbone of a polyomino. (Right) A backbone decomposed into strips (green) and corner tiles (yellow) (Color figure online).

The corner tiles will be the splitting points of the separation. The separation of the backbone can be described by three steps. In a first step, we decompose the polyomino by recursively removing the corner tiles of degree three, until only tiles with degree two are left in all components. In the second step, these components are further decomposed via the corner tiles of degree two, such that only strips remain. In a third and final step, the straight strips are decomposed, until just trivial tiles are left.

Because each corner tile has either two or three adjacent strips, the recursive separations of the backbone have a degree of at most three. The key ingredient for an efficient separation, i.e., a polylogarithmic recursion depth, for all three steps, is that the splitting points are chosen such that the sizes of the split components are balanced. In particular, for each splitting, we ensure that the size of each split component is at most the half of the size of the original component. This can be obtained by picking the respective tree median of a remaining backbone piece, i.e., at a corner whose removal leaves each connected component with at most half the number of strips of the original tree. Performing this splitting operation recursively yields a recursion depth of at most  $\mathcal{O}(\log k)$ .

Proof details can be found in the full version of the paper.

For the overall backbone assembly, we use four glues,  $\mathcal{O}(1)$  tiles and  $\mathcal{O}(k)$  bins within  $\mathcal{O}(\log n \cdot \log k)$  stages: we split at tree medians  $\mathcal{O}(\log k)$  times, and use  $\mathcal{O}(\log n)$  stages for each strip.  $\square$

By applying the approach of the backbone, we construct any polyomino by assembling its backbone and then flooding it by the set of tiles  $S$ , which is illustrated in Fig. 4. To guarantee that flooding the backbone does not exceed the original boundaries, we apply the following property of temperature-2 assemblies.

**Lemma 2.** *Consider an arbitrary supertile  $P'$  and an arbitrary set of single tiles  $S'$ , such that all non-bonded glues of  $P'$  and  $S'$  have strength 1. Each position  $p$  that is bounded (indirectly) by two parallel, non-glued sides of  $P'$  will not be part of any supertile that can be assembled by  $P'$  and  $S'$  at  $\tau = 2$ .*

*Proof.* Consider a position  $p$  that is bounded by two parallel, non-glued sides of  $P'$  and that is adjacent to two sides of  $P'$ . Then at most one of the two sides of  $P'$  can have glues. If there is a tile in  $S'$  that wants to attach there, the connection strength cannot exceed 1. Hence, the tile cannot be attached to  $P'$ . This trivially holds for positions that are adjacent to one side of  $P'$  only and for positions that are not adjacent to any side of  $P'$ . Thus, Lemma 2 holds.  $\square$

On the one hand, Lemma 2 ensures that no position outside of  $3P$  is filled by flooding the backbone. On the other hand, similar as in Theorem 1, the following simple observation guarantees that all tiles of  $3P$  that do not belong to the backbone are filled by a single tile:

*Property 1.* In the configuration of Lemma 2,  $p$  is filled by a tile in a unique self-assembly, if and only if there are sequences of collinear, adjacent tiles  $t_1$  and  $t_2$  that fulfill the following:

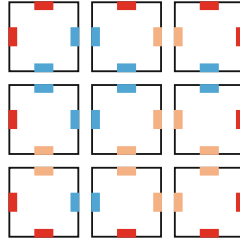
- $t_1$  and  $t_2$  are constructible from  $S'$  and  $P'$ ,
- $t_1$  and  $t_2$  meet in  $p$ , and
- $t_1$  and  $t_2$  start from perpendicular unglued sides  $f_1$  and  $f_2$  of  $P'$  that bound  $p$  (indirectly).

The flooding tiles of  $S$  are defined such that their combination with the not bonded glues of the backbone meets the properties of Lemma 2 and Property 1. Hence, by mixing the backbone and  $S$  in a single bin leads finally to a fully connected version of  $3P$ .

**Theorem 2.** *Let  $P$  be an arbitrary polyomino with  $k$  vertices. Then there is a  $\tau = 2$  staged assembly system that constructs a fully connected version of  $P$  in  $\mathcal{O}(\log^2 n)$  stages, with 7 glues,  $\mathcal{O}(1)$  tiles,  $\mathcal{O}(k)$  bins and scale factor 3.*

*Proof.* For the staged self-assembly of  $P$ , we still need to give a set of flooding tiles  $S$  and have to define how the sides of the backbone have to be marked by glues such that the flooding leads to a fully connected version of  $P$ .

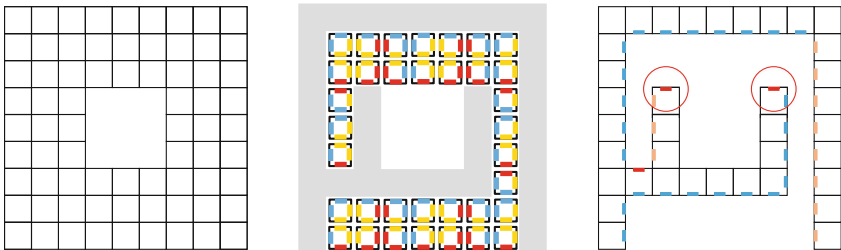
By Lemma 2 and Property 1, we know that every position that does not belong to the polyomino needs to be bounded by at least two parallel unglued sides and every tile of the polyomino must be bounded by at least two perpendicular glued sides. We can construct the backbone while satisfying these properties as follows: we cover each 3-scaled tile of  $P$  according to the glue chart, illustrated



**Fig. 4.** Glue chart for  $3 \times 3$  tiles for filling up the shape. Blue glue  $\hat{=} g_1$ , orange glue  $\hat{=} g_2$  and red glue  $\hat{=} g_3$  (Color figure online).

in Fig. 4 and mark each side of the backbone’s boundary, except for the polyomino’s and holes’ boundaries, by the glue that is induced by the glue chart (see Fig. 5, middle).

An example for a correct placements of the glues can be found in Fig. 5. Observe that there exist some strips that have a glue type on their end that is different from the glue type that is used in the strip construction (see the red circles in Fig. 5). For building those strips we have to modify the backbone assembly. We assemble those strips completely, but without the last tile with the different glue. Then we add a single tile such that the glue is on the correct place.



**Fig. 5.** (Left) A scaled polyomino with one hole. (Middle) Construction of the glues inside the backbone. (Right) Every tile of the backbone is furnished with glues. A strip may have conflicts with glues while the strip is assembling (red circle) (Color figure online).

Overall, we have four glue types for building the backbone, and three glue types for the five strip types and the connection tiles if the side points to the interior of the polyomino. Hence, we use a total of seven glue types and  $\mathcal{O}(1)$  tile types.

To fill up the polyomino, we mix the nine kinds of tiles (see Fig. 4) plus the backbone in one bin. In total, we need  $\mathcal{O}(\log^2 n)$  stages, seven glues,  $\mathcal{O}(1)$  tiles and  $\mathcal{O}(k)$  bins to assemble a fully connected polyomino, scaled by a factor 3 from the target shape.  $\square$



As noted before, the number of degree-3 corner tiles depends on the number of holes. We can describe the overall complexity in terms of  $h$ , the number of holes. For the special case of hole-free shapes, we can skip some steps, reducing the necessary number of stages. In particular, Corollary 1 follows from Theorem 2.

**Corollary 1.** *The stage complexity of Theorem 2 can be quantified in the the number of holes  $h$  such we get a stage complexity of  $O(\log^2 h + \log n)$ . In particular, Theorem 2 gives a staged self-assembly system for hole-free shapes with  $O(\log n)$  stages, seven glues,  $O(1)$  tiles,  $O(k)$  bins and a scale factor of 3.*

### 3 Fully-Connected Constructions for $\tau = 1$

In this section we describe approaches for assembling polyominoes at temperature  $\tau = 1$ .

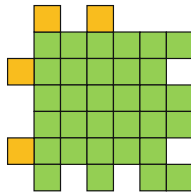
#### 3.1 Hole-Free Shapes, $\tau = 1$

We present a system for building hole-free polyominoes. The main idea is based on [4], i.e., splitting the polyomino into strips. Each of these strips gets assembled piece by piece; if there is a component that can attach to the current strip, we create it and attach it.

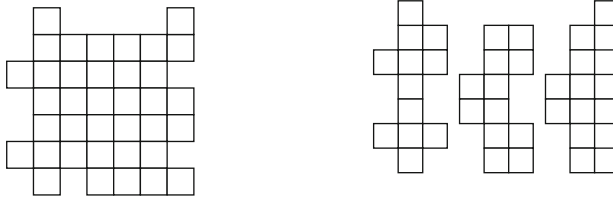
Our geometric approach partitions the polyomino into rectangles and uses them to assemble the whole polyomino. Even for complicated shapes with many vertices, this number of rectangles is never worse than quadratic in the size of the bounding box; in any case we get a large improvement in the stage complexity.

We first consider a building block, see Fig. 6.

**Lemma 3.** *A  $2n \times 2m$  rectangle (with  $n \geq m$ ) with at most two tabs at top and left side and at most two pockets at each bottom or right side (see Fig. 6) can be assembled with  $\mathcal{O}(\log n)$  stages, 9 glues,  $\mathcal{O}(1)$  tiles and  $\mathcal{O}(1)$  bins at  $\tau = 1$ .*



**Fig. 6.** A square (green) with tabs on top and left side (orange) and pockets on bottom and right side (Color figure online).



**Fig. 7.** (Left) A modified square with tabs and pockets. (Right) A partition into components.

*Proof.* First consider the  $2n \times 2m$  square, which we partition into (vertical) rectangles of width 2. As shown in Fig. 7, these are joined by tabs and pockets in rows  $n$  and  $n + 1$ . The glues on their sides are the same as for recursively cutting a square according to the jigsaw technique of [4]. Now every component has a maximum width of 3, even with the tabs. This allows us to use nine glues to create each component with attached tabs and pockets as follows:

A component without a tab or pocket is cut between the  $(n - 1)$ st and the  $n$ th row, as well as between the  $(n + 1)$ st and the  $(n + 2)$ nd row. Then we have two strips of width 2 and one  $2 \times 2$  square. The square can be assembled by brute force with desired glues on its sides. The strips can also be decomposed recursively like a  $1 \times n$  strip with desired glues on the sides. Thus, for this kind of component nine glues suffice and the component is built within  $\mathcal{O}(\log n)$  stages. Note that we need  $\mathcal{O}(1)$  bins to store every possible component of this kind, i.e., they use one out of three possible glue triples on each side.

A component with tabs and/or pockets is cut between rows, such that only components without tabs and pockets and at most four components with tabs and pockets exists. Note that the four components are either single tiles or  $1 \times 2$  strips. The other components are either strips of width two or similar to the components above. Hence, we need at most  $\mathcal{O}(\log n)$  stages to build the biggest component. Then we assemble all components by successively putting together pairs. We observe that this kind of component appears at most six times. Thus, we need six bins to store the components of this kind. Again the nine glues suffice.

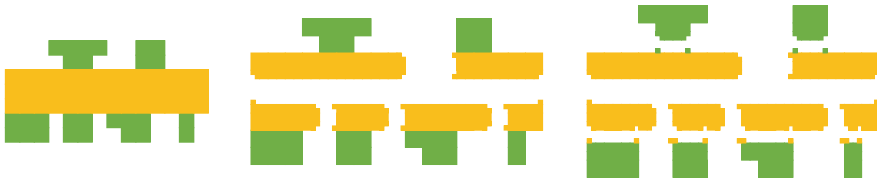
Now we have all components in  $\mathcal{O}(1)$  bins, so we can assemble the components in a pairwise fashion to the desired polyomino within  $\mathcal{O}(\log n)$  stages. Overall our nine glues suffice, so we have  $\mathcal{O}(1)$  tiles.  $\square$

**Theorem 3.** *Let  $P$  be a hole-free polyomino with  $k$  vertices. Then there is a  $\tau = 1$  staged assembly system that constructs a fully connected version of  $P$  in  $\mathcal{O}(\log^2 n)$  stages, with 18 glues,  $\mathcal{O}(1)$  tiles,  $\mathcal{O}(k)$  bins and scale factor 4.*

*Proof.* We cut the polyomino with horizontal lines, such that all cuts go through reflex vertices, leaving a set of rectangles. If  $V_r$  is the set of reflex vertices in the polyomino, we have at most  $|V_r| =: k_r$  cuts and therefore  $\mathcal{O}(k)$  rectangles. Now we find one rectangle that forms a tree median in the rectangle adjacency tree (i.e., a rectangle that splits the tree into connected components that have at

most half the number of rectangles). Recursing over this splitting operation, we get a tree decomposition of depth  $\mathcal{O}(\log k)$ . On the pieces, we use a scale factor of 2 for employing a jigsaw decomposition.

When removing a rectangle  $R$ , the remaining polyomino may be split into a number of components; see Fig. 8. To connect all of them to  $R$ , we employ another scale factor of 2, allowing us to split  $R$  in half with a horizontal line. Each half is further subdivided vertically into jigsaw components, such that each component can connect to a part of  $R$  independently from the others, as shown in the figure. When all components have been attached to some part of  $R$ , we can assemble both halves of  $R$  and then assemble these two together. Doing this for all rectangles produces  $\mathcal{O}(k_r)$  new components. Hence, our decomposition tree has at most  $\mathcal{O}(k_r) = \mathcal{O}(k)$  leaves, where the leaves are rectangles that need  $\mathcal{O}(\log n)$  for construction. This yields  $\mathcal{O}(\log k \log n)$  stages overall; the rectangle components consume  $\mathcal{O}(k)$  bins. Similar to assembling a square, we need nine glues to uniquely assemble all rectangles to the correct polyomino.



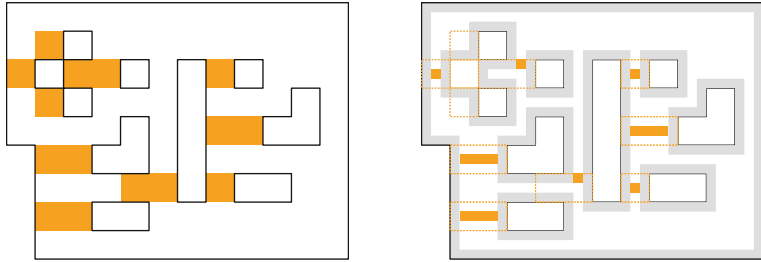
**Fig. 8.** (Left) A chosen rectangle (orange) that splits the polyomino into components (green). (Middle) Decomposition of splitting rectangle. (Right) Decomposition of the components (Color figure online).

By construction, every rectangle component has at most four adjacent rectangle components; its size is  $2w \times 2h$  for some width  $w$  and height  $h$ . The four adjacent components are all connected at different sides, so the left and upper side each have two tabs, while the right and lower side have two pockets. Thus, we can use the approach of Theorem 3 to assemble all rectangles with 9 additional glues and  $\mathcal{O}(1)$  bins for each rectangle component.

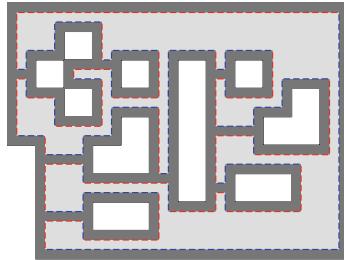
Overall, we have  $\mathcal{O}(\log n)$  stages to assemble the  $\mathcal{O}(k)$  rectangles with  $\mathcal{O}(1)$  bins for each rectangle, plus  $\mathcal{O}(\log^2 n)$  stages to assemble the polyomino from the rectangles, for a total of  $\mathcal{O}(\log^2 n)$  stages and  $\mathcal{O}(k)$  bins. For the rectangles we need nine glues, along with nine glues for the remaining assembly, for a total of 18 glues, with  $\mathcal{O}(1)$  tile types. The overall scale factor is 4.  $\square$

### 3.2 Polygons with Holes, $\tau = 1$

**Theorem 4.** *Let  $P$  be an arbitrary polyomino with  $k$  vertices. Then there is a  $\tau = 1$  staged assembly system that constructs a fully connected version of  $P$  in  $\mathcal{O}(\log^2 n)$  stages, with 20 glues,  $\mathcal{O}(1)$  tiles,  $\mathcal{O}(n)$  bins and scale factor 6.*



(a) Connections between boundaries. (b) Tunnels inside the connections.



(c) Final shapes  $S_1$  (dark) and  $S_2$  (light).

**Fig. 9.** Separation of the polyomino  $P$  into the two shapes  $S_1$  and  $S_2$ .

*Proof.* (Sketch) Complete details can be found in the full version of the paper; see Fig. 9 for the overall construction. From a high-level point of view, the approach constructs two supertiles  $S_1$  and  $S_2$  separately and finally glues them together, see Fig. 9 for the overall construction. The first supertile  $S_1$  consists of the boundaries of all holes, the boundary of the whole polyomino, and connections between these boundaries. The second supertile  $S_2$  is composed of the rest of the polyomino. A scale factor of 6 guarantees that  $S_2$  is hole-free, which in turns allows employing the approach of Theorem 3.  $\square$

## 4 Future Work

Our new methods have the same stage and bin complexity as previous work about stage assemblies [4] and use just a small number of glues. Because the bin complexity is in  $\mathcal{O}(k)$  for a polyomino with  $k$  vertices, we may need many bins if the polyomino has many vertices. Hence, all our methods are excellent for shapes with a compact geometric description. This still leaves the interesting challenge of designing a staged assembly system with similar stage, glue and tile complexity, but a better bin complexity for polyominoes with many vertices, e.g., for  $k \in \Omega(n^2)$ ?

Another interesting challenge is to develop a more efficient system for an arbitrary polyomino. Is there a staged assembly system of stage complexity  $o(\log^2 n)$  without increasing the other complexities?

## References

1. Abel, Z., Benbernou, N., Damian, M., Demaine, E.D., Demaine, M.L., Flatland, R., Kominers, S.D., Schweller, R.: Shape replication through self-assembly and rnae enzymes. In: ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1045–1064 (2010)
2. Aggarwal, G., Cheng, Q., Goldwasser, M.H., Kao, M.-Y., de Espanes, P.M., Schweller, R.T.: Complexities for generalized models of self-assembly. *SIAM J. Comput.* **34**(6), 1493–1515 (2005)
3. Cannon, S., Demaine, E.D., Demaine, M.L., Eisenstat, S., Patitz, M.J., Schweller, R.T., Summers, S.M., Winslow, A.: Two hands are better than one (up to constant factors). In: Symposium on Theoretical Aspects of Computer Science (STACS), pp. 172–184 (2013)
4. Demaine, E.D., Demaine, M.L., Fekete, S.P., Ishaque, M., Rafalin, E., Schweller, R.T., Souvaine, D.L.: Staged self-assembly: nanomanufacture of arbitrary shapes with  $o(1)$  glues. *Natural Comput.* **7**(3), 347–370 (2008)
5. Demaine, E.D., Demaine, M.L., Fekete, S.P., Patitz, M.J., Schweller, R.T., Winslow, A., Woods, D.: One tile to rule them all: simulating any tile assembly system with a single universal tile. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) *ICALP 2014*. LNCS, vol. 8572, pp. 368–379. Springer, Heidelberg (2014)
6. Fekete, S.P., Hendricks, J., Patitz, M.J., Rogers, T.A., Schweller, R.T.: Universal computation with arbitrary polyomino tiles in non-cooperative self-assembly. In: ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 148–167 (2015)
7. Fu, B., Patitz, M.J., Schweller, R.T., Sheline, R.: Self-assembly with geometric tiles. In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) *ICALP 2012, Part I*. LNCS, vol. 7391, pp. 714–725. Springer, Heidelberg (2012)
8. Padilla, J.E., Liu, W., Seeman, N.C.: Hierarchical self assembly of patterns from the robinson tilings: DNA tile design in an enhanced tile assembly model. *Natural Comput.* **11**(2), 323–338 (2012)
9. Padilla, J.E., Patitz, M.J., Schweller, R.T., Seeman, N.C., Summers, S.M., Zhong, X.: Asynchronous signal passing for tile self-assembly: fuel efficient computation and efficient assembly of shapes. *Int. J. Found. Comput. Sci.* **25**(4), 459–488 (2014)
10. Park, S.H., Pistol, C., Ahn, S.J., Reif, J.H., Lebeck, A.R., Dwyer, C., LaBean, T.H.: Finite-size, fully addressable DNA tile lattices formed by hierarchical assembly procedures. *Angewandte Chemie* **118**(5), 749–753 (2006)
11. Reif, J.H.: Local parallel biomolecular computation. *DNA-Based Comput.* **3**, 217–254 (1999)
12. Rothemund, P.W.K., Winfree, E.: The program-size complexity of self-assembled squares (extended abstract). In: ACM Symposium on Theory of Computing (STOC), pp. 459–468 (2000)
13. Somei, K., Kaneda, S., Fujii, T., Murata, S.: A microfluidic device for DNA tile self-assembly. In: *DNA Computing (DNA 11)*, pp. 325–335 (2006)
14. Winfree, E.: Algorithmic self-assembly of DNA. Ph.D thesis, California Institute of Technology (1998)