



Construction of Geometric Structure by Oritatami System

Yo-Sub Han¹ and Hwee Kim²(✉)

¹ Department of Computer Science, Yonsei University, 50 Yonsei-Ro, Seodaemun-Gu, Seoul 03722, Republic of Korea

emmous@yonsei.ac.kr

² Department of Mathematics and Statistics, University of South Florida, 12010 USF Cherry Drive, Tampa, FL 33620, USA

hweekim@mail.usf.edu

Abstract. Self-assembly is the process where smaller components autonomously assemble to form a larger and more complex structure. One of the application areas of self-assembly is engineering and production of complicated nanostructures. Recently, researchers proposed a new folding model called the oritatami model (OM) that simulates the cotranscriptional self-assembly, based on the kinetics on the final shape of folded molecules. Nanostructures in oritatami system (OS) are represented by a sequence of beads and interactions on the lattice. We propose a method to design a general OS, which we call GEOS, that constructs a given geometric structure. The main idea is to design small modular OSs, which we call hinges, for every possible pair of adjacent points in the target structure. Once a shape filling curve for the target structure is ready, we construct an appropriate primary structure that follows the curve by a sequence of hinges. We establish generalized guidelines on designing a GEOS, and propose two GEOSs.

1 Introduction

Self-assembly is the process where smaller components—usually molecules—autonomously assemble to form a larger and more complex structure. Self-assembly plays an important role in constructing biological structures and high polymers [21]. Applications of self-assembly include nanostructured electric circuits [2, 5] and smart drug delivery [13, 20] (Fig. 1).

One well-known mathematical model of the self-assembly phenomenon is the abstract tile assembly model (aTAM) by Winfree [22]. Recently, Geary et al. [7] proposed a new folding model called the oritatami model (OM) that simulates the cotranscriptional self-assembly based on the experimental RNA transcription called RNA origami [8]. In general, OM assumes that a sequence of molecules is transcribed linearly, and predicts its geometric shape from the autonomous folding of the sequence based on the reaction rate of the folding. An oritatami system (OS) consists of a sequence of beads (which is the transcript) and a set of rules for possible intermolecular reactions between beads. For each bead

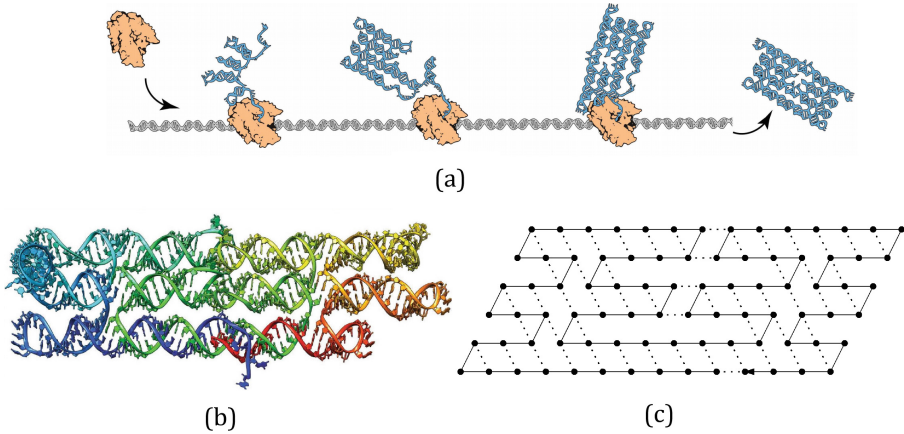
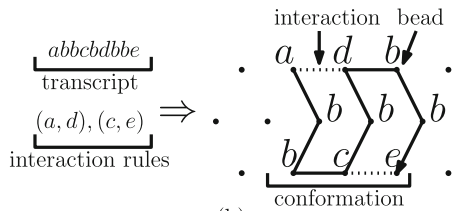


Fig. 1. The motivation of the oritatami model. (a) An illustration of an RNA Origami [7], which transcribes an RNA strand that self-assembles. (b) The product of an RNA Origami. (c) Abstraction of the product in oritatami system.

in the sequence, the system takes a lookahead of a few upcoming beads and determines the best location of the current bead that maximizes the number of possible interactions from the lookahead. Note that the lookahead represents the reaction rate of the cotranscriptional folding and the number of interactions represents the energy level (See Fig. 2 for the analogy between RNA origami and oritatami system.). Researchers designed various OSs including a binary counter [6] and a Boolean formula simulator [11]. It is known that OM is Turing complete [7] and there are several methods to optimize OSs [10, 12, 15]. There are also approaches to analyze construction of geometric structures [14, 17].

RNA Origami	Oritatami System
Nucleotides	Beads
Transcript	Sequence of beads connected by a line
h-bonds between nucleotides	Interactions
Cotranscriptional folding rate	Delay
Resulting secondary structure	Conformation

(a)



(b)

Fig. 2. (a) Analogy between RNA origami and oritatami system. (b) Visualization of oritatami system and its terms.

There are many experimental researches on engineering nanostructures using self-assembly [16, 18]. Since the trial-and-error approach in designing nanostructures is often costly, researchers instead rely on abstract models of self-assembly to engineer desired nanostructures. In aTAM, nanostructures are represented by shapes, and researchers focused on finding tile complexity of a target

shape [1, 19]. Because a nanostructure is represented by sequence of beads and bead interactions on the lattice in OS, one may ask; given a geometric structure on the lattice, how can we design an OS that constructs the given structure? A naive solution is to use unique bead types for all possible beads. However this approach is unrealistic in experiments and is not a desired solution. Instead, we want to use only a constant number of bead types and a fixed ruleset, and design a function that encodes a given geometric structure into a transcript such that the transcript folds as the given structure.

We propose a generalized method to design a geometric structure constructing OS (GEOS in short). The target structure is given as a set of points in an arbitrary lattice. We map each point in a target structure to a set of points in the triangular lattice for the OS. The main idea is to design small modular OSs, which we call hinges, for every possible pair of adjacent points in a target structure. These hinges use interactions of beads only within adjacent points instead of global interactions across many points. Moreover, the system constructs a complete structure for each point at a time instead of dividing a point into partial structures constructed at different times. This design policy yields robustness of the structure in realization. Once a shape filling curve—a skeleton sequence traversing the target structure—is given, we construct an appropriate transcript that follows the curve by a sequence of hinges (See Fig. 3). We establish generalized guidelines on designing a GEOS, and propose two GEOSs.

Recently, Demaine et al. [3] studied a similar problem of general geometric structure construction by OS. They considered a set of points on the triangular lattice, and mapped each point to an hexagon in the lattice for the OS. They filled the whole set of hexagons globally, without explicit point ordering. They proposed a basic module to fill the hexagon. They also suggested how to modify the module to be connected with neighboring modules, resulting in a target conformation spanning the whole set of hexagons. They used an OS of delay 1 and arity 4, and obtained the rigidity 1. Note that their scale is at least 19 according to our measure whereas our approach gives 25 in both designs.

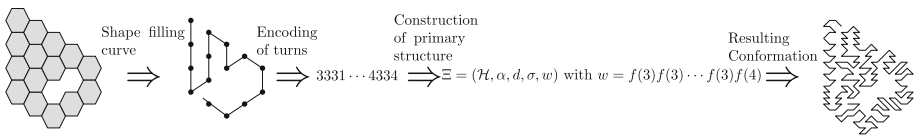


Fig. 3. An illustration of a geometric structure constructing OS. Once a shape filling curve for the given geometric structure is given, the curve is encoded as a sequence of numbers denoting consecutive turns. For each turn, we propose a partial primary structure called a hinge. The sequence of corresponding hinges forms the transcript of the geometric structure constructing OS, and the resulting conformation constructs the given structure.

2 Preliminaries

Let $w = a_1 a_2 \cdots a_n$ be a string over Σ of size n and bead types $a_1, \dots, a_n \in \Sigma$. The *length* $|w|$ of w is n . For two indices i, j with $1 \leq i \leq j \leq n$, we let $w[i, j]$ be the substring $a_i a_{i+1} \cdots a_{j-1} a_j$; we use $w[i]$ to denote $w[i, i]$. We use w^n to denote the catenation of n copies of w .

Oritatami systems operate on the triangular lattice Λ_t with the vertex set V and the edge set E . For a point p and a bead type $a \in \Sigma$, we call the pair (p, a) an *annotated point*, or simply a *point* if being annotated is clear from context. Two points p, q (or annotated points $(p, a), (q, b)$) are *adjacent* if they are at unit distance. A *path* is a sequence $P = p_1 p_2 \cdots p_n$ of *pairwise-distinct* points p_1, p_2, \dots, p_n such that $p_i p_{i+1}$ is at unit distance for all $1 \leq i < n$. Given a string $w \in \Sigma^n$, a *path annotated by w* , or simply *w -path*, is a sequence P_w of annotated points $(p_1, w[1]), \dots, (p_n, w[n])$, where $p_1 \cdots p_n$ is a path. We call points of the w -path beads, and we call the i -th point $(p_i, w[i])$ the i -th bead of the w -path. Let $\mathcal{H} \subseteq \Sigma \times \Sigma$ be a symmetric relation, specifying between which bead types can form a hydrogen-bond-based interaction (interaction for short). This relation \mathcal{H} is called the *ruleset*.

A conformation instance, or *configuration*, is a triple (P, w, H) of a directed path P in Λ_t , $w \in \Sigma^* \cup \Sigma^\omega$, and a set $H \subseteq \{(i, j) \mid 1 \leq i, i+2 \leq j, \{P[i], P[j]\} \in E\}$ of interactions. This is to be interpreted as the sequence w being folded while its i -th bead $w[i]$ is placed on the i -th point $P[i]$ along the path and there is an interaction between the i -th and j -th beads if and only if $(i, j) \in H$. Configurations (P_1, w_1, H_1) and (P_2, w_2, H_2) are *congruent* provided $w_1 = w_2$, $H_1 = H_2$, and P_1 can be transformed into P_2 by a combination of a translation, a reflection, and rotations by 60° . The set of all configurations congruent to a configuration (P, w, H) is called the *conformation* of the configuration and denoted by $C = [(P, w, H)]$. We call w a *primary structure* of C . Let \mathcal{H} be a ruleset. An interaction $(i, j) \in H$ is *valid with respect to \mathcal{H}* , or simply *\mathcal{H} -valid*, if $(w[i], w[j]) \in \mathcal{H}$. We say that a conformation C is *\mathcal{H} -valid* if all of its interactions are \mathcal{H} -valid. For an integer $\alpha \geq 1$, C is of *arity α* if the maximum number of interactions per bead is α , that is, if for any $k \geq 1$, $|\{i \mid (i, k) \in H\}| + |\{j \mid (k, j) \in H\}| \leq \alpha$ and this inequality holds as an equation for some k . By $\mathcal{C}_{\leq \alpha}$, we denote the set of all conformations of arity at most α .

Oritatami systems grow conformations by elongating them under their own ruleset. For a finite conformation C_1 , we say that a finite conformation C_2 is an *elongation* of C_1 by a bead $b \in \Sigma$ under a ruleset \mathcal{H} , written as $C_1 \xrightarrow{\mathcal{H}}_b C_2$, if there exists a configuration (P, w, H) of C_1 such that C_2 includes a configuration $(P \cdot p, w \cdot b, H \cup H')$, where $p \in V$ is a point not in P and $H' \subseteq \{(i, |P|+1) \mid 1 \leq i \leq |P| - 1, \{P[i], p\} \in E, (w[i], b) \in \mathcal{H}\}$. This operation is recursively extended to the elongation by a finite sequence of beads as follows: For any conformation C , $C \xrightarrow{\mathcal{H}^*}_\lambda C$; and for a finite sequence of beads w and a bead b , a conformation C_1 is elongated to a conformation C_2 by $w \cdot b$, written as $C_1 \xrightarrow{\mathcal{H}^*}_{w \cdot b} C_2$, if there is a conformation C' that satisfies $C_1 \xrightarrow{\mathcal{H}^*}_w C'$ and $C' \xrightarrow{\mathcal{H}}_b C_2$.

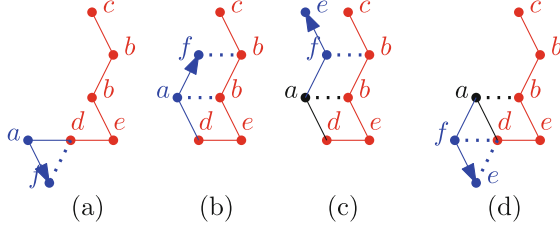


Fig. 4. An example OS with delay 2 and arity 2. The seed is colored in red, and the stabilized beads and interactions are colored in black. (Color figure online)

An *oritatami system* (*OS*) is a 6-tuple $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$, where \mathcal{H} is a *ruleset*, $\delta \geq 1$ is a *delay*, and C_σ is an \mathcal{H} -valid initial *seed* conformation of arity at most α , upon which its *transcript* $w \in \Sigma^* \cup \Sigma^{\mathbb{N}}$ is to be folded by stabilizing beads of w one at a time and minimize energy collaboratively with the succeeding $\delta - 1$ nascent beads. The energy of a conformation $C = [(P, w, H)]$ is $U(C) = -|H|$; namely, the more interactions a conformation has, the more stable it becomes. The set $\mathcal{F}(\Xi)$ of conformations *foldable* by this system is recursively defined as follows: The seed C_σ is in $\mathcal{F}(\Xi)$; and provided that an elongation C_i of C_σ by the prefix $w[1 : i]$ be foldable (i.e., $C_0 = C_\sigma$), its further elongation C_{i+1} by the next bead $w[i+1]$ is foldable if

$$C_{i+1} \in \underset{\substack{C' \in \mathcal{C}_{\leq \alpha} \text{ s.t.} \\ C_i \xrightarrow{\mathcal{H}}_{w[i+1]} C'}}{\operatorname{argmin}} \min \left\{ U(C') \mid C \xrightarrow{\mathcal{H}^*}_{w[i+2:i+k]} C', k \leq \delta, C' \in \mathcal{C}_{\leq \alpha} \right\}. \quad (1)$$

Once we have C_{i+1} , we say that the bead $w[i+1]$ and its interactions are *stabilized* according to C_{i+1} . A conformation foldable by Ξ is *terminal* if none of its elongations is foldable by Ξ .

Figure 4 illustrates an example of an OS with delay 2, arity 2 and the rule-set $\{(a, b), (b, f), (d, f), (d, e)\}$; in (a), the system tries to stabilize the first bead a of the transcript, and the elongation in (a) gives 1 interaction. However, it is not the most stable one since the elongation in (b) gives 2 interactions in total. Thus, the first bead a is stabilized according to the location in (b). In (c), the system tries to stabilize the second bead f , and the elongation in (c) gives 1 interaction for the primary structure fe . However, the elongation in (d) gives 2 interactions in total. Thus, the second bead f is stabilized according to the location in (d). Note that f is not stabilized according to the location in (b), although the elongation in (b) is used to stabilize the first bead a .

3 On the Generalized Design of GEOS

The input for the design of a geometric structure construction OS (GEOS) is as follows:

- A lattice Λ_0 on the plane.
- A shape that we want to fill on the lattice, which is given by the set \mathbb{P}_0 of points. It is necessary that the grid graph of \mathbb{P}_0 should be connected.

The output should include

- A triangular lattice Λ_t that spans Λ_0 .
- An injective function $f_c : p \in \Lambda_0 \rightarrow p_c \in \Lambda_t$ that maps a point in Λ_0 to a point in Λ_t . For each point p in Λ , we call $f_c(p)$ the *core point*.
- A bijective mapping $f_u : p \in \Lambda_0 \rightarrow \mathbb{U}(p) \subset \Lambda_t$ that maps a point in Λ_0 to a set of points in Λ_t . For each point p in Λ , we call the induced graph of $f_u(p)$ the *unit shape*. The unit shape should be a solid grid graph, and the size of $f_u(p)$ —which we call the *scale*—should be constant for all p 's. Moreover, $f_u(\Lambda_0) = \Lambda_t$. The concept of the scale is introduced while proving intrinsic universality of aTAM [4] as the size of the metatile that can simulate one tile in the system. Here, the scale represents the size of the partial primary structure that can cover one point in \mathbb{P}_0 .
- A deterministic OS $\Xi = (\Sigma, w, \mathcal{H}, \delta, \alpha, C_\sigma = [(P_\sigma, w_\sigma, H_\sigma)])$ on Λ_t , where the final conformation covers at least one point in $f_u(p)$ for each $p \in \mathbb{P}_0$.

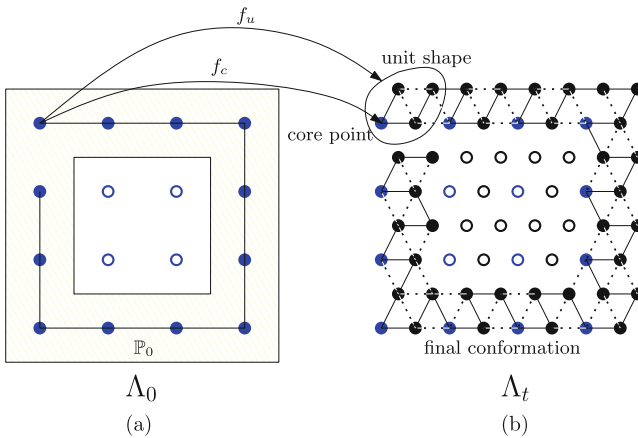


Fig. 5. An illustration of the input and the output for the GEOS. The figure in (a) shows the input and the figure in (b) shows the output. In figure (b), core points are colored in blue. The scale of the OS is 4. (Color figure online)

Figure 5 shows an example of the input and the output for the GEOS. Aside from the lattice, the mapping and OS, we establish desirable features that determine a good design of the geometric structure construction OS, motivated from the design of a shape-fitting aTAM by Soloveichik and Winfree [19].

- The scale should be as small as possible: Each point in \mathbb{P}_0 is mapped into a set of points in Λ_t , on which the conformation of the OS is stabilized. Thus, the

smaller the size of \mathbb{U} , the shorter the length of the final conformation—which helps realization of the OS in experiments.

- The number of beads in Ξ should be as small as possible: This goal is motivated from minimizing the number of tiles in an aTAM, which also helps realization of the OS in experiments.
- The final conformation should fill as many points in $f(\mathbb{P}_0)$ as possible: We use *rigidity* to refer to the lower bound of the ratio of the number of filled points to the scale.

The basic idea of the GEOS is to design small OSs for all possible pairs of unit vectors in Λ_0 . Namely, if we have a shape filling curve for \mathbb{P}_0 in Λ_0 , the curve can be represented as a sequence of unit vectors. For each point, we have an in-vector and an out-vector that represent the curve. For each pair of vectors, we design a partial OS—which we call a *hinge*—that fills adjacent unit shapes in Λ_t . We propose design guidelines that are helpful in constructing a GEOS. Although there is no need to follow all of the guidelines, following each guideline provides a necessary condition for better features described above.

1. Unit shapes should be identical. Moreover, unit shapes considering core points should have reflection and rotational symmetry. Note that the number of beads we use depends on the number of possible pairs of adjacent unit shapes. Identical and symmetric unit shapes greatly reduce the number of possible cases, as shown in Fig. 6. The unit shape in (a) has partial rotational symmetry on the triangular lattice, and there are two different types of distinct unit vectors that we should consider (Namely, there are two distinct pairs of adjacent unit shapes.). Thus, we need to design 10 different hinges. On the other hand, the unit shape in (b) has full rotational symmetry on the triangular lattice, and all pairs of adjacent unit shapes are identical. Thus, we only need to design 5 different hinges.

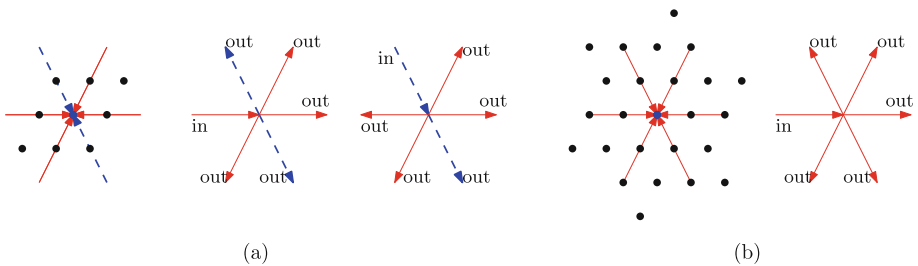


Fig. 6. Two possible unit shapes. We assume that Λ_0 is triangular.

2. We categorize beads into two categories: core beads and hinge beads. *Core beads* form a partial conformation (which we call a *core*) that covers the core point, and we use different *hinge beads* for different hinges to connect cores. We use distinct core for each pair $(\mathbb{U}(p), \vec{a})$ of a unit shape and a unit vector, which we call an *unit*. We establish two guideline for the core.

- (a) Cores for different unit vectors should be rotationally symmetric. Like the guideline for the symmetry of unit shapes, this condition greatly reduces the number of hinges.
- (b) For each point covered by the core, all neighbors of the point should be in the unit shape. Namely, core beads are not revealed on the surface of the unit shape, which prevents unintended interactions between core beads and hinge beads from another unit shape. Figure 7 shows two example cores, where only the example (a) follows the guideline.

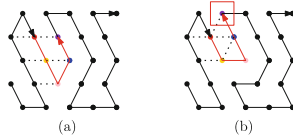


Fig. 7. Two possible cores, colored by red lines. While the core in (a) follows the guideline, the core in (b) does not, and reveals the purple core bead, which may cause unintended interactions with hinge beads from another unit shape. (Color figure online)

3. For each unit, we construct the core and the *context*—which points in the unit shape are filled before the core stabilizes. Namely, we divide $\mathbb{U}(p)$ except points occupied by core beads by two sets: preoccupied points $\mathbb{X}(p)$ and unoccupied points $\mathbb{O}(p)$. We have one more guideline for the context: For each possible pair $((\mathbb{U}(p_1), \vec{a}_1), (\mathbb{U}(p_2), \vec{a}_2))$ of adjacent units where $\vec{p}_1 + \vec{a}_2 = \vec{p}_2$, there should be a Hamiltonian path from the core of the first unit to the core of the second unit that covers $\mathbb{O}(p_1) \cup \mathbb{X}(p_2)$. This guideline ensures that the final conformation fills the maximum number of points in $f(\mathbb{P}_0)$. Figure 8 shows two example contexts, where only the example (a) follows the guideline.

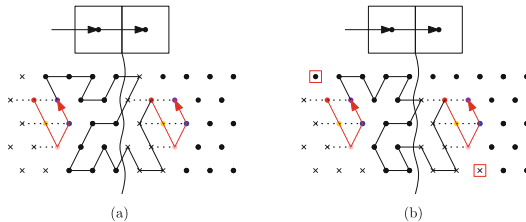


Fig. 8. Two possible contexts for units with the unit vector directing right. In each unit, preoccupied points are denoted by crosses. While the context in (a) allows maximum number of filled points, the context in (b) allows some points (in red boxes) that cannot be covered by the hinge. (Color figure online)

4. For each possible pair of adjacent units, we design a hinge that fits into the contexts. We establish one guideline for the hinge: Hinge beads interact only with hinge beads within the same hinge, or core beads. Namely, hinge beads in different types of hinges do not interact with each other. This guideline prevents unintended interactions between different types of hinges, and we only need to check interactions with the adjacent, same type of the hinge in the validation process. Figure 9 shows two example hinges, where only the example (a) follows the guideline.

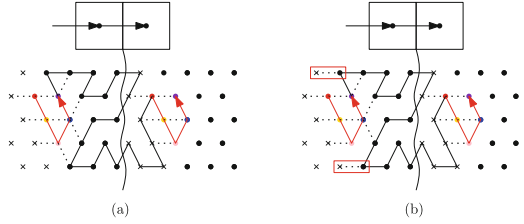


Fig. 9. Two possible hinges for the same pair of adjacent units. While the hinge in (a) has no interaction between current hinge beads and beads in preoccupied points, the hinge in (b) has some (in red boxes), which may cause unintended interaction when the neighboring units are filled with hinge beads in preoccupied points. (Color figure online)

Table 1. Summary of strengths of four guidelines on desirable features of GEOS.

Guidelines	Small $ \Sigma $	High rigidity	Avoiding unintended interactions
(i)	✓		
(ii)(a)	✓		
(ii)(b)			✓
(iii)		✓	
(iv)			✓

We summarize the strengths of four guidelines on desirable features of GEOS in Table 1. Following the proposed design guidelines, we design two GEOSs in the following section.

4 Two GEOS Designs

4.1 A GEOS Oriented from a Triangular Lattice

In the first GEOS that we call Ξ_{Δ} , we set A_0 as a triangular lattice. Note that we use a shape filling curve to encode the given geometric structure in the

triangular lattice, and not all connected triangular grid graph is Hamiltonian. Thus, in general we assume that a shape filling curve instead of a geometric shape on the lattice is given as an input. However, when \mathbb{P}_0 is a triangular lattice, we can construct a Hamiltonian path in return for tripling the scale. In the mapping of \mathbb{P}_0 to $\mathbb{U}(\mathbb{P}_0)$, we group and map three unit shapes and core points for one point in \mathbb{P}_0 to use the algorithm proposed by Gordon et al. [9], who proved that there exists a polynomial algorithm to find a Hamiltonian path in a connected, locally connected triangular grid graph. We can successfully make the grid graph of $\mathbb{U}(\mathbb{P}_0)$ locally connected by mapping multiple core points and adding additional filling points to $\mathbb{U}(\mathbb{P}_0)$ as shown in Fig. 10.

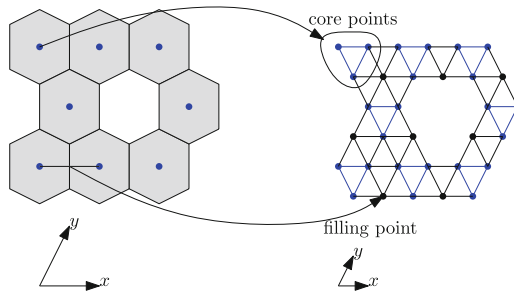


Fig. 10. Construction of a locally connected grid graph from \mathbb{P}_0 . We map three core points from a point in \mathbb{P}_0 . In addition, we add a filling point between two sets of core points for each edge in the grid graph of \mathbb{P}_0 . The resulting grid graph is always locally connected.

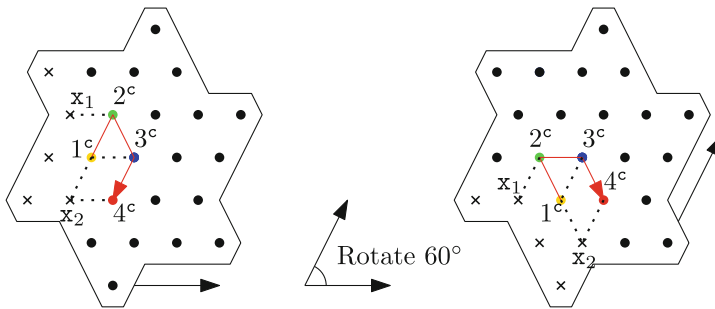


Fig. 11. Two of six units including cores. Cores are represented with colored beads, dotted lines and red arrowed lines. Crosses represent preoccupied points. All units and cores are rotationally symmetric according to unit vectors. (Color figure online)

Formally, the set \mathbb{P}_0 of points on the lattice A_0 is encoded to the set \mathbb{C}_t of core points on the triangular lattice A_t as follows:

- For each point $(x, y) \in \mathbb{P}_0$, we add three points $(2x, 2y), (2x - 1, 2y + 1), (2x, 2y + 1)$ to \mathbb{C}_t .
- If two points $(x, y), (x + 1, y)$ are in \mathbb{P}_0 , we add $(2x + 1, 2y)$ to \mathbb{C}_t .
- If two points $(x, y), (x, y + 1)$ are in \mathbb{P}_0 , we add $(2x - 1, 2y + 2)$ to \mathbb{C}_t .
- If two points $(x, y), (x - 1, y + 1)$ are in \mathbb{P}_0 , we add $(2x + 1, 2y + 2)$ to \mathbb{C}_t .

Note that the grid graph of \mathbb{C}_t is always locally connected.

In Ξ_Δ , we use one unit shape and six unit vectors (from Λ_0). A part of the units including cores is shown in Fig. 11. We use integers to represent bead types and superscripts to represent different sets of bead types. We observe that all units and cores are rotationally symmetric according to unit vectors, following design guidelines (i) and (ii)(a). Thus, we need to consider only 5 hinges. We can observe that cores are not revealed on the surface of unit shapes, following design guideline (ii)(b). Note that we use distinct sets of beads for different hinges, and there is no interaction between hinge beads from different hinges, following design guideline (iv). Also note that for each hinge, there exists one point that cannot be filled. Thus, the rigidity of this OS is $24/25$.

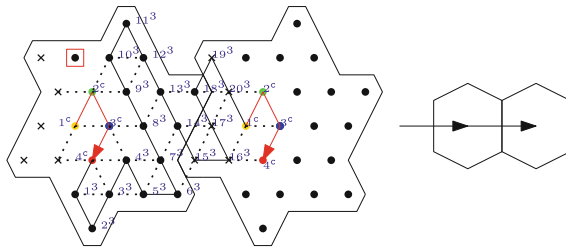


Fig. 12. An illustration of the hinge h_1 . Core beads are represented with superscript c , and hinge beads are represented with superscript 1.

Figure 12 shows one of the five hinges, representing that the shape filling curve proceeds straight. We use 20 distinct hinge beads for this hinge. The delay is 4, and each bead in the hinge is stabilized by at least 4 interactions. Note that once all hinges are given, it is straightforward to design sequences of beads for borderlines of the shape filling curve, where the starting sequence becomes a seed and the ending sequence becomes a suffix of the primary structure.

An example of Ξ_Δ is shown in Fig. 13. Once the set \mathbb{P} of points is given as in (a), we construct the set of core points as in (b) and find a Hamiltonian path for the grid graph. Then we connect hinges according to the triples of consecutive points in the path as in (c). Red lines represent core beads.

4.2 A GEOS Oriented from a Square Lattice

Note that the rigidity of Ξ_Δ is not 1. We design the second GEOS called Ξ_\square that uses Λ_0 as a square lattice, whose rigidity is 1. Since not all connected grid

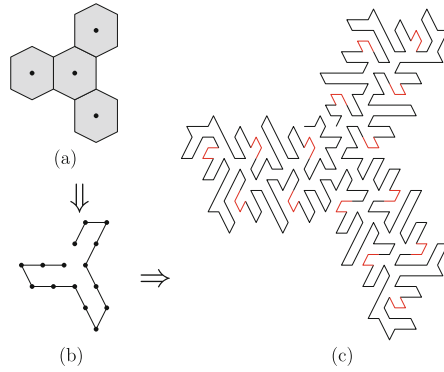


Fig. 13. An example of Ξ_{Δ}

graph of Λ_0 is Hamiltonian, we first transform a square lattice into an affine triangular lattice, as in Fig. 14(b). Then, we can transform the grid graph into a locally connected grid graph by quadrupling the scale as in Fig. 14(c). From the locally connected grid graph, we can extract a Hamiltonian path as in Fig. 14(d).

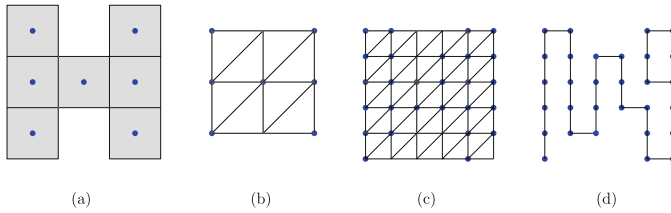


Fig. 14. (a) The input shape (b) the input shape on the affine triangular lattice (c) construction of a locally connected graph (d) a retrieved Hamiltonian path

We use two unit shapes and four unit vectors (from Σ_0). Figure 15 shows the mapping of core points and unit shapes from Λ_0 . Two types α, β of unit shapes of size 25 appear in Fig. 15, where one is rotationally symmetric to the other.

Units including cores are shown in Fig. 16. We refer to a hinge as a pair of an unit shape type and an unit vector, i.e. (α, up) . Considering symmetry of units, we need to design 11 hinges. The delay of the system is 5, and each bead in the hinge is stabilized by at least 5 interactions. We can observe the following properties of the design of hinges.

- All neighbors of the core points are in the unit shape, following design guideline (ii)(b).
- Hinge beads in different types of hinges do not interact with each other, following design guideline (iv).

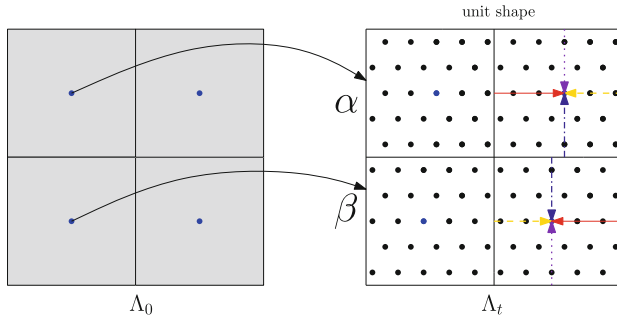


Fig. 15. The mapping of core points and unit shapes from Λ_0 . Arrows on the right unit shapes represent unit vectors

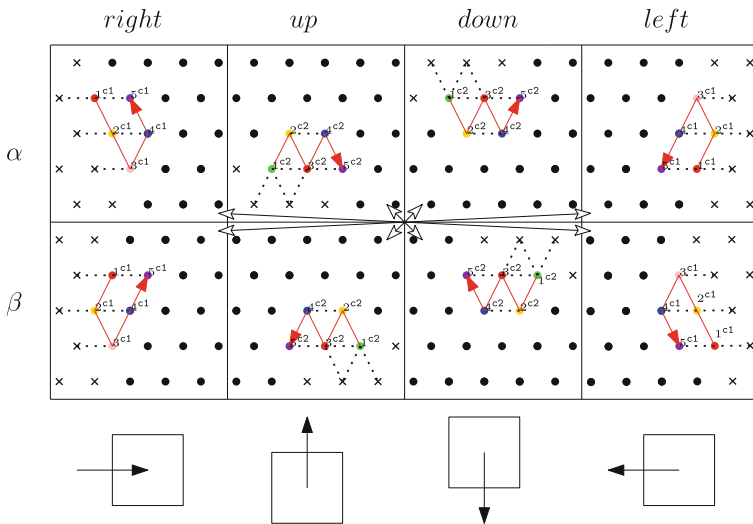


Fig. 16. Eight units including cores. Cores are represented with colored beads, dotted lines and red arrowed lines. Crosses represent preoccupied points. Rotationally symmetric units are paired by white arrows. (Color figure online)

- In all hinges that connect two points p_1 and p_2 , the grid graph of $\mathbb{O}(p_1) \cup \mathbb{X}(p_2)$ is Hamiltonian, following design guideline (iii). Moreover, every hinge fills $\mathbb{O}(p_1) \cup \mathbb{X}(p_2)$. Thus, Ξ_{\square} covers all points in $f(\mathbb{P}_0)$.
- There are hinges covering at least 2 pairs of units due to the rotational symmetry of units.
- When the filling curve goes upward, we consider only one hinge due to the reflection symmetry of cores in units (α, up) and (β, up) .
- In the hinge connecting two units (β, up) and $(\beta, right)$, the core for $(\beta, right)$ is horizontally rotated. From the point in the shape filling curve that maps

to the unit, we regard all hinges as horizontally rotated until we use another hinge for (β, up) and $(\beta, right)$.

An example of \mathbb{P}_0 in a square lattice is shown in the left of Fig. 17. Given the set of points \mathbb{P}_0 , the GEOS fills points in $f(\mathbb{P}_0)$ as in the right of Fig. 17.

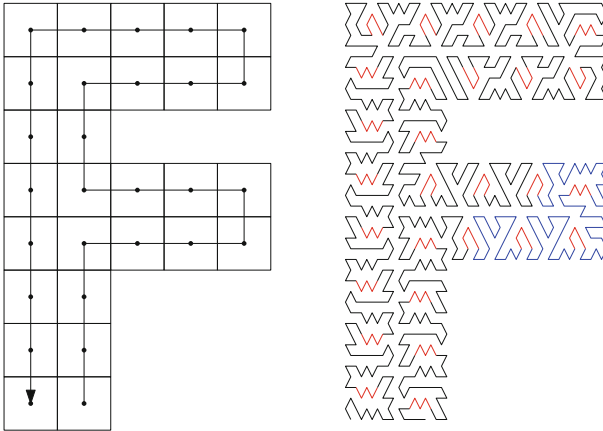


Fig. 17. (Left) An example of \mathbb{P}_0 in a square lattice. A Hamiltonian path that covers the grid graph of \mathbb{P}_0 is denoted by the arrowed line. (Right) The final conformation that fills $f(\mathbb{P}_0)$. The conformation starts and ends at the downmost unit shapes. Red lines represent cores, black lines represent hinges and blue lines represent horizontally rotated hinges. (Color figure online)

5 Conclusions

We have established generalized guidelines on designing a GEOS, and proposed two GEOSs summarized in Fig. 18. Although we can construct an arbitrary shape out of a GEOS, there are optimization problems. In general, reducing the scale also reduces the number of hinge beads we use, but also makes it harder to design symmetric unit shapes for the tessellation and may increase the number of units to consider. Reducing the scale may also increase the possibility of unintended interaction between different hinges. Thus, it is a challenging question to find a GEOS that uses the minimum number of bead types while achieving maximum rigidity.

GEOS	Scale	Rigidity	# of Hinges	# of Bead Types
Ξ_{Δ}	25 (75 when group core points)	24/25	5	104
Ξ_{\square}	25 (100 when group core points)	1	11	223

Fig. 18. Summary of two proposed GEOSs

Our two GEOS designs have approached the objective by setting up four design guidelines. It turns out that the first Ξ_{Δ} cannot achieve rigidity 1, while the second Ξ_{\square} achieves rigidity 1 using more bead types. Our future work includes finding bead complexity—the lower bound of the number of beads—for a given lattice A_0 and optimization of GEOSs.

Acknowledgements. This work has been supported in part by the NIH grant R01 GM109459.

References

1. Adleman, L., Cheng, Q., Goel, A., Huang, M.-D., Wasserman, H.: Linear self-assemblies: equilibria, entropy, and convergence rates. In: Proceedings of the 6th International Conference on Difference Equations and Applications, pp. 51–60 (2001)
2. Bhuvana, T., Smith, K.C., Fisher, T.S., Kulkarni, G.U.: Self-assembled CNT circuits with ohmic contacts using Pd hexadecanethiolate as in situ solder. *Nanoscale* **1**(2), 271–275 (2009)
3. Demaine, E.D., et al.: Know when to fold 'em: self-assembly of shapes by folding in oritatami. In: Doty, D., Dietz, H. (eds.) DNA 2018. LNCS, vol. 11145, pp. 19–36. Springer, Cham (2018)
4. Doty, D., Lutz, J.H., Patitz, M.J., Schweller, R.T., Summers, S.M., Woods, D.: The tile assembly model is intrinsically universal. In: Proceedings of the IEEE 53rd Annual Symposium on Foundations of Computer Science, pp. 302–310 (2012)
5. Eichen, Y., Braun, E., Sivan, U., Ben-Yoseph, G.: Self-assembly of nanoelectronic components and circuits using biological templates. *Acta Polym.* **49**(10–11), 663–670 (1998)
6. Geary, C., Meunier, P., Schabanel, N., Seki, S.: Efficient universal computation by greedy molecular folding. CoRR, abs/1508.00510 (2015)
7. Geary, C., Meunier, P., Schabanel, N., Seki, S.: Programming biomolecules that fold greedily during transcription. In: Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science, pp. 43:1–43:14 (2016)
8. Geary, C., Rothmund, P.W.K., Andersen, E.S.: A single-stranded architecture for cotranscriptional folding of RNA nanostructures. *Science* **345**, 799–804 (2014)
9. Gordon, V.S., Orlovich, Y.L., Werner, F.: Hamiltonian properties of triangular grid graphs. *Discret. Math.* **308**(24), 6166–6188 (2008)
10. Han, Y.-S., Kim, H.: Ruleset optimization on isomorphic oritatami systems. In: Brijder, R., Qian, L. (eds.) DNA 2017. LNCS, vol. 10467, pp. 33–45. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66799-7_3
11. Han, Y., Kim, H., Ota, M., Seki, S.: Nondeterministic seedless oritatami systems and hardness of testing their equivalence. *Nat. Comput.* **17**(1), 67–79 (2018)
12. Han, Y.-S., Kim, H., Rogers, T.A., Seki, S.: Self-attraction removal from oritatami systems. In: Pighizzini, G., Câmpeanu, C. (eds.) DCFS 2017. LNCS, vol. 10316, pp. 164–176. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60252-3_13
13. Li, J., Fan, C., Pei, H., Shi, J., Huang, Q.: Smart drug delivery nanocarriers with self-assembled DNA nanostructures. *Adv. Mater.* **25**(32), 4386–4396 (2013)
14. Masuda, Y., Seki, S., Ubukata, Y.: Towards the algorithmic molecular self-assembly of fractals by cotranscriptional folding. In: Câmpeanu, C. (ed.) CIAA 2018. LNCS, vol. 10977, pp. 261–273. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94812-6_22

15. Ota, M., Seki, S.: Rule set design problems for oritatami systems. *Theor. Comput. Sci.* **671**, 26–35 (2017)
16. Pistol, C., Lebeck, A.R., Dwyer, C.: Design automation for DNA self-assembled nanostructures. In: *Proceedings of the 43rd ACM/IEEE Design Automation Conference*, pp. 919–924 (2006)
17. Rogers, T.A., Seki, S.: Oritatami system; a survey and the impossibility of simple simulation at small delays. *Fundamenta Informaticae* **154**(1–4), 359–372 (2017)
18. Santis, E.D., Ryadnov, M.G.: Self-assembling peptide motifs for nanostructure design and applications. *Amino Acids Peptides Proteins* **40**, 199–238 (2016)
19. Soloveichik, D., Winfree, E.: Complexity of self-assembled shapes. *SIAM J. Comput.* **36**(6), 1544–1569 (2007)
20. Verma, G., Hassan, P.A.: Self assembled materials: design strategies and drug delivery perspectives. *Phys. Chem. Chem. Phys.* **15**(40), 17016–17028 (2013)
21. Whitesides, G.M., Boncheva, M.: Beyond molecules: self-assembly of mesoscopic and macroscopic components. *Proc. Natl. Acad. Sci. U.S.A.* **99**(8), 4769–4774 (2002)
22. Winfree, E.: Algorithmic self-assembly of DNA. Ph.D. thesis, California Institute of Technology (1998)