Iterative Self-Assembly with Dynamic Strength Transformation and Temperature Control

Dandan Mo and Darko Stefanovic

Department of Computer Science, University of New Mexico,
MSC01 1130 1 University of New Mexico, Albuquerque, NM 87131-0001, U.S.A
`{mdd,darko}@cs.unm.edu`

Abstract. We propose an iterative approach to constructing regular shapes by self-assembly. Unlike previous approaches, which construct a shape in one go, our approach constructs a final shape by alternating the steps of assembling and disassembling, increasing the size of the shape iteratively. This approach is embedded into an extended hexagonal tile assembly system, with dynamic *strength transformation* and *temperature control*. We present the construction of equilateral triangles as an example and prove the uniqueness of the final shape. The tile complexity of this approach is $O(1)$.

Keywords: Algorithmic self-assembly, hexagonal tiles, strength transformation, temperature control

1 Introduction

The *tile assembly system* (TAS) contains a self-assembly process in which small tiles autonomously attach to a seed, assembling into a larger and more complex shape. TAS dates back to the late 1990s. In 1998, Winfree proposed a mathematical model of DNA self-assembly [1] with the operations of *Annealing*, *Ligation*, and *Denaturation*. The same year, Winfree et al. presented a tile assembly model [2] which connected tiling theory with structural DNA nanotechnology, using the double-crossover molecules (DX) to implement the model. In 2000, Rothemund and Winfree proposed the *abstract tile assembly model* (aTAM) [3], in which the tiles are squares with sticky ends on each side. Adleman et al. extended this model with stochastic *time complexity* [4].

Later works concerning the aTAM or TAS have been mainly extensions aimed at improving the complexity or exploring more final shapes that can be self-assembled. Instead of the original square tiles, triangular, hexagonal [5, 6], and string tiles [7] can also implement the TAS. Aggarwal et al. extended the standard TAS by allowing the assembly of the super tiles [8]. Demaine et al. proposed a staged self-assembly model, where the tiles can be added in sequence [9].The tile complexity problem usually studies the cost of building an $N \times N$ square in terms of the number of tile types required. Compared with the previous result N^2 [3] at temperature[1] 1, geometric tiles [10] use only $\Theta(\sqrt{\log N})$ tile types. An extended model with a mechanism of temperature programming [11, 12] uses only $O(1)$ tile types. Besides an $N \times N$ square, other regular

---

[1] The temperature is a threshold value that determines if a shape is stable.

shapes can be self-assembled using the TAS. Cook et al. showed that a Sierpinski triangle can be formed by 7 different square tiles [13]. Kari et al. presented a triangular tile self-assembly system [14] where an equilateral triangle of size N can be formed using N^2 tile types at temperature 1, and $2N-1$ tile types at temperature 2. Woods et al. proposed an active self-assembly model [15] to construct algorithmic shapes and patterns in polylogarithmic time.

Here we propose an iterative approach to self-assembly, with a tile complexity of $O(1)$. The approach extends a tile assembly system with dynamic *strength transformation* and *temperature control*. Unlike previous TAS models, where the final stable shape is formed in one go, our approach constructs regular shapes of increasing size in each iteration in a geometric progression. This iterative process consists of assembling an n-size shape from an n-size seed cluster[2] and disassembling it into two pieces which then form a new seed cluster with a larger size for the next iteration. In other words, our approach alternates the steps of *assembling* and *disassembling*. We demonstrate how this approach works on the example of constructing equilateral triangles; the rest of the paper treats a specific tile assembly system for triangles.

In the following, we first give an overview of our iterative approach in Section 2. In Section 3, a formal definition of the extended hexagonal TAS is given. In Section 4, we present a concrete example of enlarging an equilateral triangle from size four to size six[3]. In Section 5, we prove that an equilateral triangle is the only final stable shape that can be self-assembled within this system.

2 Overview of the Approach

Our iterative approach repeats two steps: assembling and disassembling. In the assembling step, we start from a seed cluster, which is one of the three sides of the triangle when an iteration ends. The free individual tiles present in solution attach to this seed cluster, updating the shape until an equilateral triangle is completed. The assembling step takes place under a constant temperature τ_1, which is also the threshold value to determine if a shape is stable[4]. The disassembling step begins when we raise the temperature to τ_2, which will make the τ_1-stable shape become unstable under τ_2, and lead to disassembly. The original shape then disassembles into two smaller shapes that are τ_2-stable. We then lower the temperature down to τ_3 to make the two shapes form a new seed cluster with larger size. When we change the temperature from τ_3 back to τ_1, the next iteration starts.

The assembling and disassembling step are enabled by two new features added to the hexagonal TAS. *Strength transformation* changes some of the bond strengths of the τ_1-stable shape and two strand strengths, in preparation for the disassembling step. *Temperature control* implements the switch between the two steps. These two features are supported by the **Rule** set and the **Operation** set in the extended hexagonal TAS, which we will discuss in the next section.

---

[2] A seed cluster of size n is a line consisting of n hexagonal tiles.

[3] The size of an equilateral triangle is denoted as its edge length.

[4] If a shape does not disassemble under temperature τ, it is τ-stable.

3 Formal Definition of the Extended Hexagonal TAS

In this section, we give a formal expression of the extended hexagonal TAS, with detailed explanations of each term. We then introduce the set of tile types in this system and illustrate the whole procedure of our iterative approach. Finally, we consider plausible chemical implementations of some operations included in the *strength transformation*.

3.1 A Formal Expression

The extended hexagonal TAS is formally a tuple $M = \{\Sigma, S, B, \Omega, U, R, O\}$, where Σ is a temperature set, S is a strand set, B is a bond set, Ω is a tile set, U is a stable shape set, R is a rule set, and O is an operation set.

Temperature Set. $\Sigma = \{\tau | \tau \in \mathbb{Z}_{>0}\}$ is the set of the temperatures.
In our example of constructing an equilateral triangle, there are three threshold temperatures in this system. $\Sigma = \{\tau_1 = 3, \tau_2 = 7, \tau_3 = 5\}$.

Strand Set And Bond Set $S = \{x | x \in A\} \cup \{\bar{x} | x \in A\}$ is the set of DNA strand types, where A denotes a symbol set. Two complementary strands x and $\bar{x}$ can form a bond with the corresponding strength. We use *bond*(x,$\bar{x}$ $\|$ w) to represent a bond and its sticking strength w, where $w \in \mathbb{Z}_{>0}$. B is the set of bonds. Thus, for all $b \in B$, we have $b =$*bond*(x,$\bar{x}$$\|$$w$) such that $x, \bar{x} \in S$. Here, w is the bond strength as well as the strength of strand x and $\bar{x}$.
In our example, $S = \{a, y, \gamma, Y\} \cup \{\bar{a}, \bar{y}, \bar{\gamma}, \bar{Y}\}$. Table 1 shows all the strands we use. For example, $a, \bar{a}$ form a bond with strength 1.

Tile Set. Ω is the set of tiles. Each tile $t = (s_1, s_2, s_3 ..., s_6)$ is abstracted as a hexagon, each side represents a strand. An example is shown in Figure 1. In our graphical representation, if a side does not have any strand, we use the dotted line to represent it. A pair of complementary strands, e.g., a and $\bar{a}$, are represented by a single solid line and a double solid line respectively. The number labeled next to the strand indicates the strand strength.The sequence of $s_i (1 \leq i \leq 6) \in S \cup \{\perp\}$ represents each side of the hexagon from the top right clockwise to the top left. A side without any strand is denoted as $\perp$. Tiles are allowed to rotate, but not to flip. Therefore, if there are two tiles $t_1 = (s_1, s_2 ..., s_6)$, $t_2 = (s'_1, s'_2 ..., s'_6)$ and if $\exists i, j$ $(1 \leq i, j \leq 6$ and $j = 7 - i)$ such that $(s'_i, s'_{i+1} ..., s'_6) = (s_1, s_2 ..., s_j)$ and $(s'_1, s'_2 ..., s'_{i-1}) = (s_{j+1}, s_{j+2} ..., s_6)$, we say that t_1 and t_2 are the same tile. We assume that the strands of the tiles are fixed in their positions, thus the angles between each two are fixed as well.

Stable Shape Set. U is the set of stable shapes. 1. $\forall t \in \Omega$, $t \in U$; 2. each shape $T \in U$ is either $T = \{t\} \cup T'$ or $T = T_1 \cup T_2$, where $t \in \Omega$ and T', T_1, T_2 are also stable shapes. That is, a stable shape is a single tile or a set of tiles that are grouped together because of the bonds between them. A stable shape can be formed in two ways—either a single tile

Table 1. Strand strength

Strand type x	a, $\bar{a}$	y, $\bar{y}$	γ, $\bar{\gamma}$	Y, $\bar{Y}$
Strand strength	1	2	5	7

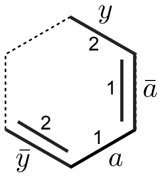

Fig. 1. This figure shows an example of a hexagon tile: $t = (y, \bar{a}, a, \bar{y}, \perp, \perp)$. The numbers labeled next to the lines indicates the strengths of the strands.

attaches to an existing stable shape or two stable shapes attach to each other. Both ways are based on the premise that the sum of the interacting strengths equals or exceeds the current temperature, which is expressed as the "assembling" rule that will be discussed shortly.

Rule Set. $R = \{assembling, stability - checking, disassembling\}$ is a set of rules that guide this system, shown in Figure 2.

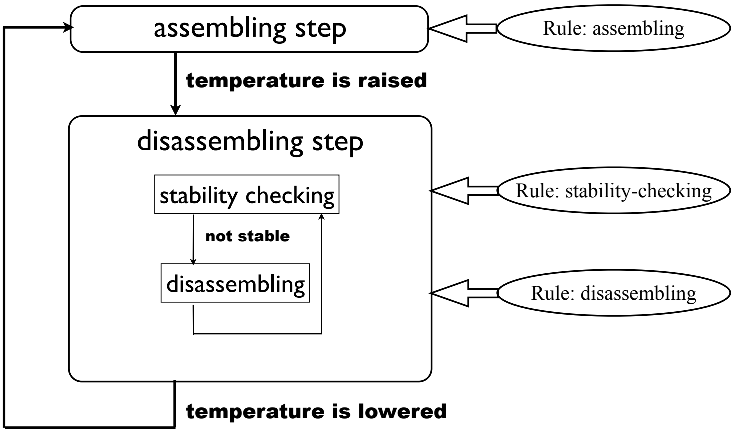

Fig. 2. How the rules guide the steps of assembling and disassembling

If the temperature is not changed, the assembling rule is applied. Whenever the temperature is raised, the stability-checking rule and the disassembling rule are applied to execute the disassembling step. We use the stability-checking rule to check the stability of the shape. If the shape is stable, there is no change to the shape, otherwise the disassembling rule is applied to break the shape. Whenever the temperature is lowered, the assembling rule is applied.

Rule 1: Assembling. The assembling rule guides the procedure of shape formation. For $T = \{t\} \cup T'$ under temperature τ, where T' is an existing stable shape, the interaction strength sum E between t and T' must be $E \geq \tau$, otherwise T will not be stable. Similarly, for $T = T_1 \cup T_2$ where T_1, T_2 are stable, the interaction strength sum E between T_1 and T_2 must be $E \geq \tau$ as well, otherwise shape T will be disassembled into the two smaller but stable shapes T_1 and T_2. Under the assembling rule, the formed shape is guaranteed to be stable.

Rule 2: Stability-Checking. The stability-checking rule is applied to check the stability of a shape after the temperature is raised. There are two levels of checking. Suppose that we have a shape T and the temperature is raised to τ. We first check if each tile $t \in T$ is stable in the shape, then check if every two sub-shapes T_1, T_2 are stably attached to each other($T_1 \cup T_2 = T$) if the first-level stability is not achieved. In the first level, for every tile $t \in T$, we compare the sum of the interaction strength between t and $T - t$ with τ_2. If the sum $< \tau$, T is not stable. Otherwise, we proceed to the second-level checking. In the second level, if there exists a cut that splits T into two sub-shapes T_1, T_2 and the sum of interaction strengths between T_1 and T_2 is less than τ, T is not stable. Only when both levels of checking succeed, can we say that shape T is stable.

Rule 3: Disassembling. The disassembling rule is applied when the shape is not stable. It works together with the stability-checking rule to disassemble the shape into smaller stable shapes. There are two levels of disassembling. The first level makes certain single tiles fall off from the shape; the second level splits the shape into two new shapes. In the first-level disassembling, we execute the first-level stability checking to find the tiles whose interaction strengths are the weakest among those tiles with interaction strength less than the current temperature, and remove these tiles from the shape. In the second-level disassembling, we execute the second-level stability checking to find the cut that needs the minimum strengths, and split the shape into two new shapes. The first-level and the second-level disassembling cannot happen at the same time, the one that needs minimum strengths takes place first. If the strengths needed are the same, the first-level disassembling comes first. The stability-checking rule is applied to the new shapes after either level disassembling is executed. If the shapes are not stable, the disassembling rule is applied to the current shapes, otherwise the disassembling step ends.

Operation Set. $O = O_1 \cup O_2 \cup \{End\}$ is a set of operations allowed in the system. Set O_1 includes the operations related to the strength transformation. These operations need the help of some specific restriction enzymes, ligase and auxiliary tiles. Operation $Op_{i \to j} \in O_1 (i, j \in \mathbb{Z}_{>0})$ changes the strength from i to j. Set O_2 includes the operations related to the temperature control, operation $\tau_i \to \tau_j \in O_2 (\tau_i, \tau_j \in \Sigma)$ changes the temperature from τ_i to τ_j. The last set contains only one operation. When End is executed, the construction procedure ends. These operations are executed when the current shapes are all stable, in other words, during an assembling step or a disassembling step, the experimentalist should not execute any operation.

In our example, $O = \{Op_{5\to2}, ligation, Op_{2\to5}\} \cup \{\tau_1 \to \tau_2, \tau_2 \to \tau_3, \tau_3 \to \tau_1\} \cup \{End\}$. $Op_{5\to2}$ changes a bond from strength 5 to strength 2(with Enzyme I, Enzyme II), $Op_{2\to5}$ changes a strand from strength 2 to strength 5(with Enzyme III and the $Op_{2\to5}$ helper). The operation of $ligation$ needs the help of a ligase, changing a bond from strength 2 to strength 7. Enzyme I, Enzyme II and Enzyme III are restriction enzymes, which will be discussed with details in Section 3.4. The $Op_{2\to5}$ helper is explained in Section 3.2.

3.2 Tile Types

In the system of constructing an equilateral triangle, there are four categories of tiles: initial component tiles, free tiles, functional tiles, and variant tiles. The first three are directly made manually; the variant tiles result from reactions with enzymes. Initial component tiles form a seed cluster of size four; it is the first stable shape existing in the solution before any reaction begins. Free tiles remain in the solution all the time. Functional tiles come in two flavors, one ($Op_{2\to5}$ helper) is included in the operation $Op_{2\to5}$, while the other (ending tile) will end the whole procedure with an equilateral triangle. The functional tiles are added to the solution by the experimentalist at certain time points. Variant tiles only appear when one of the operations in $\{Op_{5\to2}, ligation, Op_{2\to5}\}$ is executed. The initial component tiles, free tiles and functional tiles are shown in Figure 3. We will explain the transformations to the variant tiles in Section 3.4, together with the implementations of some operations. In Figure 3, y_{adp} and $\bar{y}_{adp}$ are two special strands on $Op_{2\to5}$ helper tile. Strands y_{adp} and $\bar{y}$ can form a special bond, and likewise the strands $\bar{y}_{adp}$ and y.

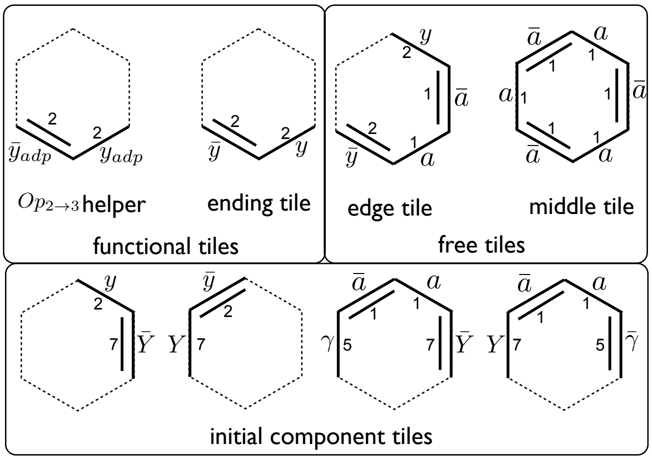

Fig. 3. Tile types is shown in this figure. The dotted lines represent “no strand”. Each strand is represented by a single or double solid line with a name label and a strength label.

3.3 The Procedure of the Iterative Approach

The assembling step and the disassembling step together constitute one iteration of the construction. There are two parameters associated with each iteration. One is the seed cluster, which is one edge of the triangle at the beginning of each iteration; the other is the current temperature τ. In each iteration, the assembling step starts with $\tau = 3$ and a seed cluster, and ends with an equilateral triangle with the last piece left; the disassembling step starts with $\tau = 7$ and ends with two symmetric shapes that will form the new seed cluster at $\tau = 5$. When $\tau = 3$, the next iteration starts. Table 2 summarizes the actions during an iteration. The "Action" column is the operation executed at each step, the "State" column expresses the current condition of the shape and temperature. Notice that at step 1 we have two branches — choosing the *End* operation will end the iteration with an equilateral triangle of size l, otherwise we proceed to the next step. At step 2, bond$(y, \bar{y} \| 2)$ is transformed to bond$(Y, \bar{Y} \| 7)$. At step 3, bond$(\gamma, \bar{\gamma} \| 5)$ is broken into strands y and $\bar{y}$ that will form bond$(y, \bar{y} \| 2)$ later.

Table 2. The procedure of an iteration and how to stop the iteration.

Step No.	Action	State
0		seed cluster of size l, $\tau = 3$
1		an equilateral triangle with the last piece left
	End	iteration ends with an equilateral triangle of size l
2	*ligation*	bond$(y, \bar{y} \| 2) \to$ bond$(Y, \bar{Y} \| 7)$
3	$Op_{5 \to 2}$	bond$(\gamma, \bar{\gamma} \| 5) \to$ y and $\bar{y} \to$ bond$(y, \bar{y} \| 2)$
4	$Op_{2 \to 5}$	$y \to \gamma$, $\bar{y} \to \bar{\gamma}$
5	$\tau_1 \to \tau_2$	two symmetric shapes, $\tau = 7$
6	$\tau_2 \to \tau_3$	new seed cluster of size $l = 2l - 2$ is formed, $\tau = 5$
7	$\tau_3 \to \tau_1$	go to step 0

3.4 Proposed Implementation of the Operations

Figure 4 and Figure 5 show some plausible implementations of operations $Op_{5 \to 2}$, $Op_{2 \to 5}$, and *ligation*. For each implementation, we provide a high-level picture on the top to show how that operation changes the tiles. Tiles to the right of the arrow are the variant tiles. Below each high-level picture are the details concerning the strands.

In Figure 4, the left column shows the steps of transforming strand $\bar{y}$ into strand $\bar{\gamma}$. The operation orders are important. Enzyme III must be added after the tile $Op_{2 \to 5}$ helper attaches to the shape, otherwise it might be difficult to tell if $\bar{\gamma}$ is formed. Enzyme III is a restriction enzyme which is assumed to recognize the specific sequences on B-B' part. When the helper tile attaches to the shape, by observing the shape, we are sure that the intermediate product shown in (2) is formed, which will be transformed into $\bar{\gamma}$ after the cutting in (3). If we add Enzyme III before the helper tile fills in its position, the sequence 2 and 5 might be cut off from the helper tile so that the helper tile cannot attach to the shape. In this case, whether the intermediate product is formed is hard to know, so it is hard to know if strand $\bar{\gamma}$ is formed. The right column follows the similar steps, transforming strand y into strand γ.

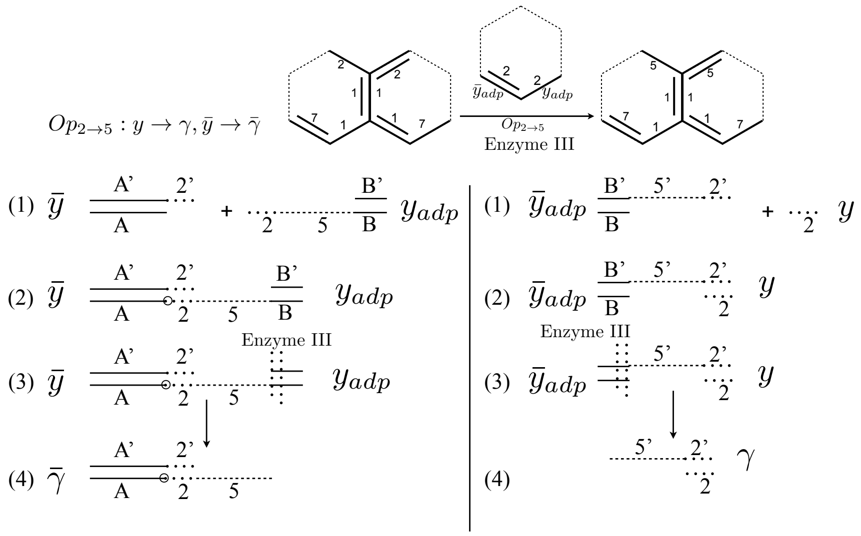

Fig. 4. This figure shows the plausible implementation of operation $Op_{2\to5}$. Letters A, B and numbers 2, 5 indicate different sequences. A', B', 2' and 5' are the corresponding complementary sequences. There are two sides for each strand, the side with the strand name is fixed on the tile, the other side is free. For example, in (1) of the left column, the left ends of A' and A are fixed, the right end of 2' is free.The left column shows the steps of transforming strand $\bar{y}$ into strand $\bar{\gamma}$. The 2' end on $\bar{y}$ sticks to the 2 end on $\bar{y}_{adp}$, forming (2). Enzyme III is a restriction enzyme which is assumed to recognize the specific sequences on B-B' part. It can cut (2) at the position indicated by the two vertical dotted lines, transforming $\bar{y}$ into $\bar{\gamma}$, which is shown in (3) and (4). The small circle between the sequence A and 2 means no connection. The right column follows the similar steps, transforming strand y into strand γ.

In Figure 5, the left part is the implementation of operation $Op_{5\to2}$, the right part is the implementation of operation *ligation*. In the implementation of $Op_{5\to2}$, two restriction enzymes are used to cut the sequences at two different positions, which is shown in (2). After the cutting, bond($\gamma, \bar{\gamma} \| 5$) is broken into two strand $y, \bar{y}$ and sequence fragments (sequence 2, 5 and sequence 5', 2). Strand y and $\bar{y}$ form a new bond later, which is shown in (3). The sequence fragments leave the tiles. In the implementation of *ligation*, ligase is used to connect the sequence A and 2, transforming bond($y, \bar{y} \| 2$) into bond($Y, \bar{Y} \| 7$).

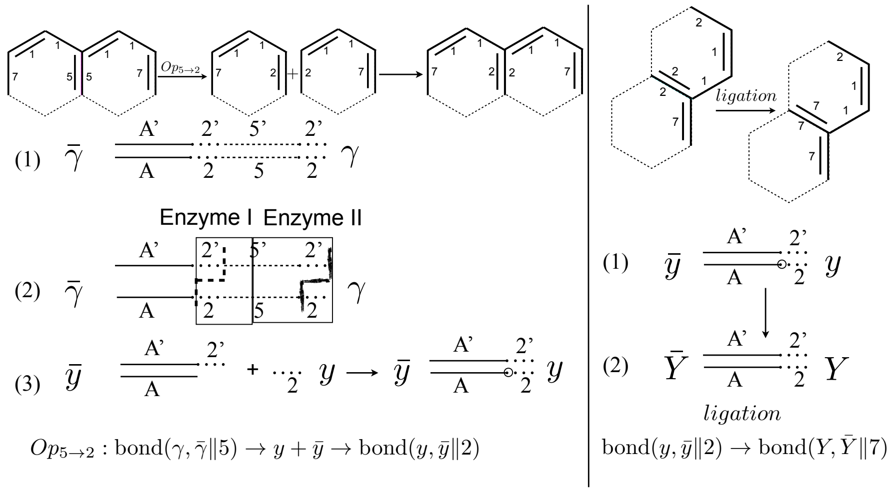

Fig. 5. This figure shows the plausible implementations of operation $Op_{5\to2}$ and *ligation*. **(a)** The operation $Op_{5\to2}$ is implemented by cutting twice. (1) shows the bond formed by γ and $\bar{\gamma}$. In (2), there are two restriction enzymes. We assume that Enzyme I recognizes the sequences within the left square, and it can cut the bond as indicated by the dashed line. We assume that Enzyme II recognizes the sequences within the right square, and it can cut the bond as indicated by the thick line. After the cutting, what are left on the tiles are the strands $\bar{y}$ and y, which can form a new bond($y, \bar{y}\|2$) later. **(b)** In the part of *ligation*, the small circle between two sequences means no connection. In (1), the bond formed by y and $\bar{y}$ has strength 2. This strength is proportional to the length of their interacting part, which is 2-2' here. After ligation, parts A and 2 are connected, which means that the interacting part between the two strands is A-A' plus 2-2' now. Since this is longer, the strength of the bond is stronger. We define the new bond formed by A-A' and 2-2' has strength 7, and use bond($Y, \bar{Y}\|7$) to represent it.

4 Constructing An Equilateral Triangle from Size 4 to 6

Figure 6 and Figure 7 show the complete procedure of constructing an equilateral triangle of size 6 from one of size 4. The number labeled next to the hexagon side indicates the strength of the strand that side represents. The strands with strength 1 are not labeled for convenience in the figures. The initial state is a seed cluster of size 4 and temperature $\tau = \tau_1 = 3$. While the temperature is not changed, the edge tiles and middle tiles in solution attach to the seed cluster one by one, constructing a stable shape shown in state 1. The operation *ligation* transforms the bonds of strength 2 into the bonds of strength 7, updating the shape to state 2. We then execute the $Op_{5 \to 2}$ operation, which needs the help of Enzyme I and Enzyme II. After that, the shape is updated to state 3, where the bond of strength 5 is transformed into a bond of strength 2. At that point, operation $Op_{2 \to 5}$ is executed. This operation has two phases: in phase (a), the helper tile fills in the top position of the shape, forming two special bonds (the bond consisting of y_{adp}, y and the bond consisting of $\bar{y}_{adp}, \bar{y}$); in phase (b), two restriction enzymes are added into the solution. They find the two special bonds, cut them at a specific position, which transforms y (single solid line with 2) into γ (single solid line with 5) and $\bar{y}$ (double solid line with 2) into $\bar{\gamma}$ (double solid line with 5), updating the shape to state 4. When we raise the temperature from τ_1 to τ_2, the disassembling step begins. The middle tile falls off from the shape first, since its total interaction strengths to the shape sum to 6, less than the current temperature $\tau_2 = 7$. The shape is then disassembled from the middle into two symmetric shapes as shown in state 5, since cutting the shape from the middle only needs strength 3, which is less than τ_2. When we change the temperature from τ_2 to τ_3, two free ends labeled as 5 can form a bond of strength 5, which constructs a new seed cluster as shown in state 6. Finally, we change the temperature to τ_1 again to start the next iteration. In Figure 7, the second iteration starts with a seed cluster of size 6, which is shown in state 0. Putting the ending tile into the solution ends the construction with an equilateral triangle as shown in state 1. The stable final shape is an equilateral triangle of side 6.

5 Equilateral Triangle is the Only Stable Final Shape

To prove that within this extended TAS the equilateral triangle is the only stable final shape, we need to prove (1) during the assembling step, no other shape except for an equilateral triangle is formed; (2) after the disassembling, there are only two shapes in solution; (3) at temperature 5, the two shapes formed after the disassembling step will bond together to form the new seed cluster for the next iteration.

(1) During the assembling step, $\tau = 3$, no two free tiles can bond together, since the strands on them have strength of 2 at most. There is only one stable shape in solution, which is the seed cluster, so the only way that a new shape is formed is that one free tile attaches to the seed cluster, updating the shape, then comes another free tile. For the edge tile, it can only stick to the leftmost position or the rightmost position of the seed cluster. For the middle tile, it can only stick to the seed cluster with the help of one of its neighbors. That is, the assembling procedure is level-by-level and from-edge-to-center. In addition to the angle design, the only stable shape formed is the equilateral triangle.

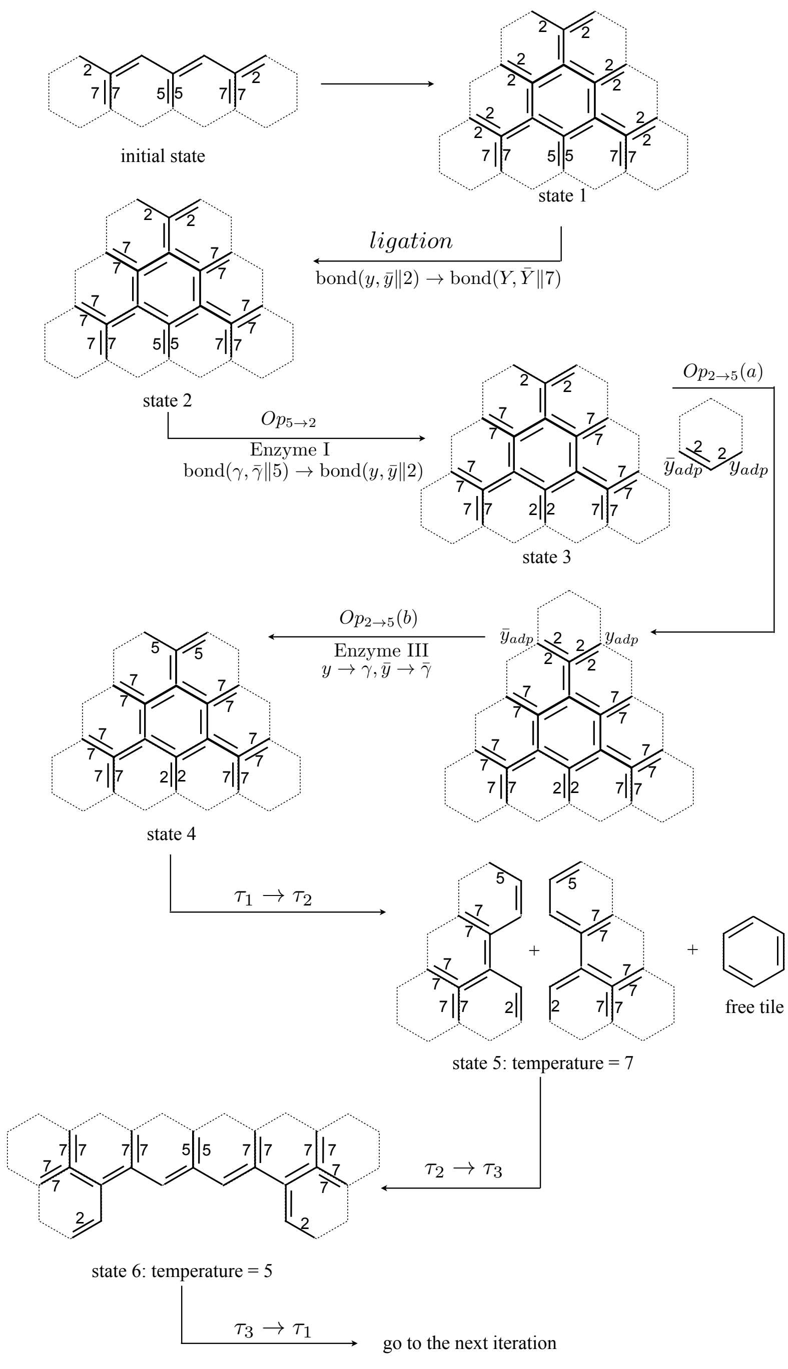

Fig. 6. Constructing an equilateral triangle of side 6 from 4: the first iteration

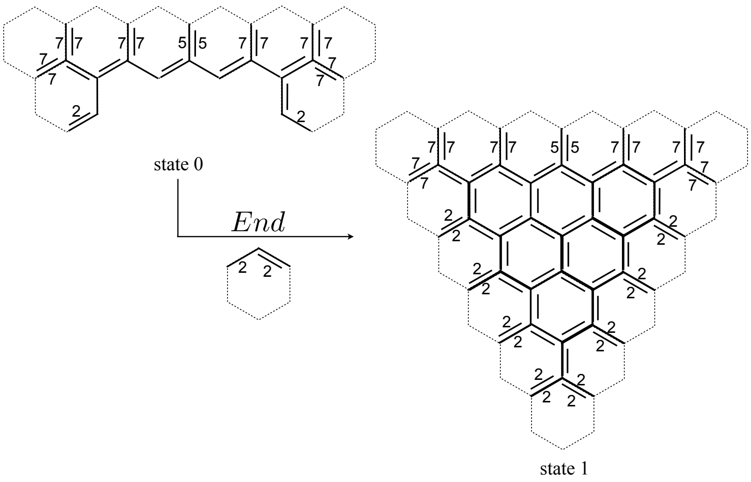

Fig. 7. Constructing an equilateral triangle of side 6 from 4: the second iteration

(2) When $\tau = 7$, the middle tiles in the one-piece-left triangle start to fall off first since they have the smallest sum of interaction strengths. After that, two bonds $b_1 = bond(a,\bar{a}\|1)$, $b_2 = bond(y,y\|2)$ are broken, which splits the current shape into two stable shapes.

(3) When the temperature $\tau = 5$, the free strands that can stick together are γ and $\bar{\gamma}$. Because the two shapes each have one of them, bond$(\gamma,\bar{\gamma}\|5)$ will be formed, which groups the two shapes together to form a new seed cluster.

6 Conclusion and Future Work

The iterative approach we propose in this paper uses only $O(1)$ tile types to construct an equilateral triangle of size n, where $n = 2^i + 2$ (i is an integer and $i \geq 1$). This approach approximately doubles the size of the cluster after each iteration, which means we can start from a small-size seed cluster instead of needing a large one proportional to the final size.

A future direction is to generalize this approach to construct other kinds of shapes, e.g., a square. The iterative characteristic is supported by the dynamic *strength transformation* and *temperature control*. Since we only need to switch between three temperatures, it is not hard to control the temperature in lab in this case. We also propose an implementation of the *strength transformation* by using conceptual enzymes, a ligase and specific restriction enzymes, as a blueprint for realizing the system in the laboratory which we hope to do.

Acknowledgments. This material is based upon work supported by the National Science Foundation under grants 1027877 and 1028238.

References

1. Winfree, E.: Algorithmic Self-Assembly of DNA. **1998** (1998)
2. Winfree, E., Liu, F., Wenzler, L.A., Seeman, N.C.: Design and self-assembly of two-dimensional DNA crystals. Nature **394**(6693) (August 1998) 539–44
3. Rothemund, P.W.K., Winfree, E.: The program-size complexity of self-assembled squares (extended abstract). In: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of computing. STOC ’00, New York, NY, USA, ACM (2000) 459–468
4. Adleman, L., Cheng, Q., Goel, A., Huang, M.D.: Running time and program size for self-assembled squares. Proceedings of the Thirty-Third Annual ACM Symposium on Theory of computing - STOC ’01 (2001) 740–748
5. Chelyapov, N., Brun, Y., Gopalkrishnan, M., Reishus, D., Shaw, B., Adleman, L.: DNA Triangles and Self-Assembled Hexagonal Tilings. Journal of the American Chemical Society **126**(43) (2004) 13924–13925
6. Kari, L., Seki, S., Xu, Z.: Triangular and hexagonal tile self-assembly systems. In Dinneen, M., Khoussainov, B., Nies, A., eds.: Computation, Physics and Beyond. Volume 7160 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 357–375
7. Winfree, E., Eng, T., Rozenberg, G.: String tile models for DNA computing by self-assembly. In Condon, A., Rozenberg, G., eds.: DNA Computing. Volume 2054 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2001) 63–88
8. Aggarwal, G., Cheng, Q., Goldwasser, M., Kao, M., de Espanes, P., Schweller, R.: Complexities for generalized models of self-assembly. SIAM Journal on Computing **34**(6) (2005) 1493–1515
9. Demaine, E., Demaine, M., Fekete, S., Ishaque, M., Rafalin, E., Schweller, R., Souvaine, D.: Staged self-assembly: nanomanufacture of arbitrary shapes with o(1) glues. Natural Computing **7**(3) (2008) 347–370
10. Fu, B., Patitz, M., Schweller, R., Sheline, R.: Self-assembly with geometric tiles. In Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R., eds.: Automata, Languages, and Programming. Volume 7391 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 714–725
11. Kao, M.Y., Schweller, R.: Reducing tile complexity for self-assembly through temperature programming. Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete algorithm - SODA ’06 (2006) 571–580
12. Summers, S.: Reducing tile complexity for the self-assembly of scaled shapes through temperature programming. Algorithmica **63**(1-2) (2012) 117–136
13. Cook, M., Rothemund, P., Winfree, E.: Self-assembled circuit patterns. In Chen, J., Reif, J., eds.: DNA Computing. Volume 2943 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2004) 91–107
14. Kari, L., Seki, S., Xu, Z.: Triangular tile self-assembly systems. In Sakakibara, Y., Mi, Y., eds.: DNA Computing and Molecular Programming. Volume 6518 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 89–99
15. Woods, D., Chen, H.L., Goodfriend, S., Dabby, N., Winfree, E., Yin, P.: Active self-assembly of algorithmic shapes and patterns in polylogarithmic time. In: Proceedings of the 4th conference on Innovations in Theoretical Computer Science. ITCS ’13, New York, NY, USA, ACM (2013) 353–354