The Complexity of Fixed-Height Patterned Tile Self-Assembly*

Shinnosuke Seki[1]** and Andrew Winslow[2]

[1] The University of Electro-Communications, Tokyo, Japan
s.seki@uec.ac.jp
[2] Université Libre de Bruxelles, Brussels, Belgium
awinslow@ulb.ac.be

Abstract. We characterize the complexity of the PATS problem for patterns of fixed height and color count in variants of the model where seed glues are either chosen or fixed and identical (so-called *non-uniform* and *uniform* variants). We prove that both variants are NP-complete for patterns of height 2 or more and admit $O(n)$-time algorithms for patterns of height 1. We also prove that if the height and number of colors in the pattern is fixed, the non-uniform variant admits a $O(n)$-time algorithm while the uniform variant remains NP-complete. The NP-completeness results use a new reduction from a constrained version of a problem on finite state transducers.

Keywords: tile self-assembly, DNA computing, finite state transducer

1 Introduction

Winfree [13] introduced the *abstract tile assembly model (aTAM)* to capture nanoscale systems of DNA-based particles aggregating to form intricate crystals, leading to an entire field devoted to understanding the theoretical limits of such systems (see surveys by Doty [3] and Patitz [10]). Ma and Lombardi [9] introduced the *patterned self-assembly tile set synthesis (*PATS*)* problem, of designing a tile set of minimum size that assembles into a given $n \times h$ colored pattern by attaching to an L-shaped seed.

Czeizler and Popa [2] were the first to provide a proof that the PATS problem is NP-hard, thus establishing the problem as NP-complete. Subsequent work studied the hardness of the constrained version where the patterns have at most c colors, called the c-PATS problem. This line of work proved the 60-PATS [11], 29-PATS [6], 11-PATS [7], and finally the 2-PATS [8] problems NP-complete.

Here we study the complexity of parameterized *height-h* PATS and c-PATS problems where patterns have a specified fixed height h and increasing width

* A full version of this paper can be found at http://arxiv.org/abs/1604.07190

** Work supported in part by JST Program to Disseminate Tenure Tracking System, MEXT, Japan, No. 6F36 and by JSPS Grant-in-Aid for Research Activity Start-up No. 15H06212.

n. We consider both *uniform* and *non-uniform* model variants, where the glues along the seed are fixed and identical or chosen in tandem with the tile set, respectively. We characterize the computational complexity of these problems via the following results:

- The height-2 PATS problem is NP-complete in both models (Sec. 4).
- The uniform height-2 3-PATS problem is NP-complete (Sec. 5).
- The non-uniform height-h c-PATS problem and uniform height-1 PATS problems admit $c^{c^{O(h)}}n$-time and $O(n)$-time algorithms, respectively (Sec. 6).

The NP-completeness results also apply to patterns of height greater than 2. Thus the complexity of the PATS problem for all combinations of height, color, and uniformity are characterized, except uniform height-2 2-PATS.

The NP-hardness reductions are based on a reduction for a new variant of the minimum-state finite state transducer problem, originally proved NP-hard by Angluin [1] and by Vazirani and Vazirani [12]. In this variant, any solution transducer is also promised to satisfy additional constraints on its transitions. The reduction is also substantially simpler than the reduction given in [1] and uses input and output strings of just two symbols, rather than the three of [12].

2 Preliminaries

Patterns, tiles, assemblies, and seeds. Define $\mathbb{N}_k = \{1, 2, \ldots, k\}$. A *pattern* is a partial function $P : \mathbb{N}^2 \to C$, i.e. a function that maps a rectangular region of lattice points to a set of colors. If $\mathrm{dom}(P) = \mathbb{N}_w \times \mathbb{N}_h$, then P is a *width-w height-h* pattern. The codomain of P, i.e. the colors seen in the pattern, is denoted $\mathrm{color}(P)$. A pattern P is *c-color* provided $|\mathrm{color}(P)| \leq c$.

A *tile type* t is a colored unit square with each edge labeled; these labels are called *glues*. A tile type's color is denoted $\mathrm{color}(t)$. For a direction $d \in \{\mathtt{N}, \mathtt{W}, \mathtt{S}, \mathtt{E}\}$, $t[d]$ denotes the glue assigned to side d of t. A tile type is non-rotatable, and thus is uniquely identified by its color and four glues. Instances of tile types, called *tiles*, are placed with their centers in $\mathbb{N}^2$.

An *assembly* is an arrangement of tiles from a set of tile types T; formally a partial function $A : \mathbb{N}^2 \to T \cup \{\varnothing\}$. A *seed* is an "L-shaped" assembly with domain $\{(0, 0)\} \cup \{(x, 0) : x \in \mathbb{N}_w\} \cup \{(0, y) : y \in \mathbb{N}_h\}$ for some $w, h \in \mathbb{N}$. The *pattern* of an assembly A is defined as $P_A((x, y)) = \mathrm{color}(A((x, y)))$ for $(x, y) \in \mathrm{dom}(A) \cap \mathbb{N}^2$, i.e. the color pattern of A, *excluding* the seed.

RTASs. A *rectilinear tile assembly system (RTAS)* is a pair $\mathcal{T} = (T, \sigma)$, where T is a set of tile types and σ is a *seed*. An assembly A *yields* an assembly A' with $\mathrm{dom}(A') = \mathrm{dom}(A) \cup \{(x, y)\}$ provided $(x-1, y), (x, y-1) \in \mathrm{dom}(A)$ and $A((x-1, y))[\mathtt{E}] = A'((x, y))[\mathtt{W}]$, $A((x, y-1))[\mathtt{N}] = A'((x, y))[\mathtt{S}]$. The set of *producible assemblies* of an RTAS are those that can be yielded, starting with the seed assembly σ. That is:

RTAS Tiling Rule: A tile of type t can be added to an assembly A at location (x, y) provided $(x-1, y), (x, y-1) \in \mathrm{dom}(A)$ and the east and north glues of the tiles at $(x-1, y)$ and $(x, y-1)$ are the same as the west and south glues of t, respectively.

As a result, tiling proceeds from southwest to northeast, i.e., a tile is first placed at $(1,1)$, then at either $(1,2)$ or $(2,1)$, etc. The *terminal assemblies* of a RTAS are the producible assemblies that do not yield other (larger) assemblies. If every terminal assembly of the system has pattern P, the system is said to *uniquely self-assemble* P. An RTAS (T, σ) is *directed*, i.e. deterministic, provided that for any distinct tile types $t_1, t_2 \in T$, either $t_1[\mathtt{W}] \neq t_2[\mathtt{W}]$ or $t_1[\mathtt{S}] \neq t_2[\mathtt{S}]$.

Uniform RTASs. We also define a practical variant of a RTAS called a *uniform* RTAS. An RTAS (T, σ) is *uniform* provided there exist two glues $\ell_{\mathtt{E}}, \ell_{\mathtt{N}}$ such that $\sigma((x,0))[\mathtt{E}] = \ell_{\mathtt{E}}$ for all $x \in \mathbb{N}_w$ and $\sigma((0,y))[\mathtt{N}] = \ell_{\mathtt{N}}$ for all $y \in \mathbb{N}_h$. In other words, the seed glues cannot be programmed and are generic.

The PATS problem. The *pattern self-assembly tile set synthesis problem (PATS)* [9] asks for the minimum-size RTAS that uniquely self-assembles a given rectangular color pattern, where the size of an RTAS (T, σ) is $|T|$, the number of tile types. Bounding the number of colors or height of the input pattern yields the following practically motivated special cases of PATS:

Problem 1 (c-colored PATS *or c-*PATS*).* Given a c-colored pattern P and integer t, does there exist an RTAS of size $\leq t$ that uniquely self-assembles P?

Problem 2 (Height-h PATS*).* Given a height-h pattern P and integer t, does there exist an RTAS of size $\leq t$ that uniquely self-assembles P?

Restricting the system to be uniform gives rise to *uniform* variants as well, contrasting with the conventional *non-uniform* variants.

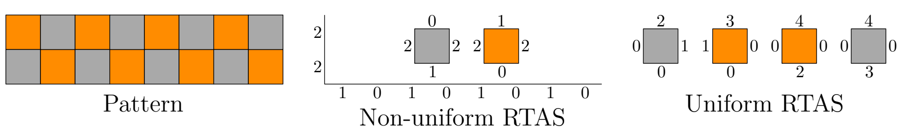


Fig. 1. A height-2 2-color pattern and minimum-size RTASs uniquely assembling the pattern in the non-uniform and uniform models.

3 Minimum-State Finite State Transducer is NP-hard

The reduction is from 3-PARTITION, a well-known NP-hard problem on integers, to a problem on *finite state transducers* or *FSTs*: finite automata where each transition is augmented with an output symbol and thus *transduces* an input string into an output string of equal length.

*Problem 3 (*3-PARTITION*).* Given a multiset of integers[3] $A = \{a_1, a_2, \ldots, a_{3n}\}$ with $\sum_{a_i \in A} a_i/n = p$ and $p/4 < a_i < p/2$, does there exist a partition of A into n sets, each with sum p?

[3] 3-PARTITION is *strongly* NP-hard, meaning that the problem is NP-hard when the elements of A are given in unary.

Theorem 1 ([4]). 3-PARTITION *is* NP-*hard.*

Formally, a FST is a 4-tuple $T = \langle \Sigma, Q, s_0, \delta \rangle$, where Σ is the *alphabet*, Q is a finite set of *states* of T, $s_0 \in Q$ is the *start state* of T, and $\delta : Q \times \Sigma \to Q \times \Sigma$ is the *transition function* of T. The *size* of T is equal to $|Q|$. An input-output quadruple $\delta(s_i, b) = (s_j, b')$ is a *transition*, i.e., a (b, b')-*transition* or b-*transition.*

*Problem 4 (*ENCODING BY FST*).* Given two strings S, S' and integer K, does there exist a FST with at most K states that transduces S to S'?

Lemma 1. ENCODING BY FST *is* NP-*hard.*

Proof. We borrow from [12] the approach of constructing S and S' by concatenating *segments*: pairs of input and output substrings of equal length that enforce specific structure in a solution FST. An input string A and output string B paired as a segment is denoted $A \to B$.

The integer output by the reduction is $K = 3pn + n + 1$, where n is the number of parts in the partition and p the size of each part. The first segment is $0^{K-1}00^{K-1} \to 0^{K-1}10^{K-1}$. This segment enforces that a solution FST must have K states; label them $s_1, s_2, \ldots, s_K$. Then for all $i < n$, $\delta(s_i, 0) = (s_{i+1}, 0)$ and $\delta(s_K, 0) = (s_1, 1)$.

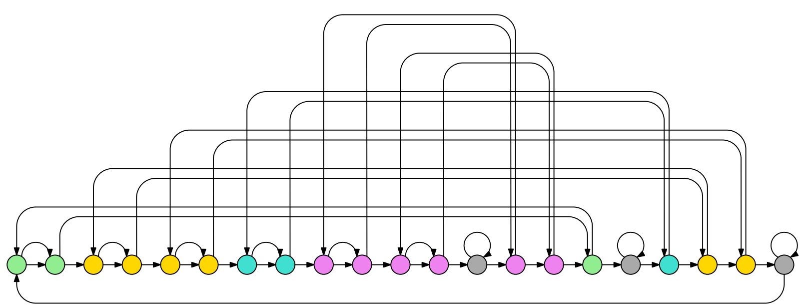


Fig. 2. A solution FST for a toy reduction from an (invalid) 3-PARTITION instance with integers $1, 2, 1, 2$ to ENCODING BY FST. The left-to-right states are s_1 to s_K, colored by their half-fixed interval (fixed singletons are gray). Transitions above the states are $(1, 1)$-transitions. All others are $(0, 0)$-transitions except the lowermost, a $(0, 1)$-transition.

The problem of partitioning integers of A into groups of size p is implemented in the collection of 1-transitions that leave each state. Each state has a 1-transition that either points to itself (a *fixed singleton*) or is one edge in a 3-cycle formed by two consecutive specified states and an unspecified third state (a *half-fixed triple*). Half-fixed triples are further organized into *half-fixed intervals*, each consisting of a group of $2a_i$ consecutive specified states and a group of a_i consecutive unspecified states for some distinct a_i. The states are partitioned into three groups:

- States s_1 through s_{2pn} are the specified halves of the half-fixed intervals.
- $n+1$ equally-spaced fixed singletons in states $s_{2pn+1}, \ldots, s_K$.
- The remaining pn states in states $s_{2pn+1}, \ldots, s_K$ partitioned into n sets of p consecutive states.

See Figure 2 for a toy example of the reduction.

The unspecified halves of the half-fixed intervals can be assigned to the third group of states if and only if the input 3-PARTITION instance has a solution. All that remains is to describe the segments that force the construction of a fixed singleton, half-fixed triple, and half-fixed interval.

Fixed singleton. The fixed singleton segment ensures that a given state s_i has $\delta(s_i, 1) = (s_i, 1)$. This is done by moving the current state to s_i, transducing a 1 to a 1, and checking whether the current state is still s_i (see Figure 3).

$0^{i-1}10^{K-i}0 \rightarrow$
$0^{i-1}10^{K-i}1$

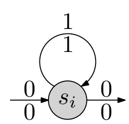


Fig. 3. The fixed single segment and corresponding FST structure enforced.

Half-fixed triple. The half-fixed triple segment forces two specified *fixed* states s_i, s_{i+1} and an unspecified *free* third state s_j to have $\delta(s_i, 1) = (s_{i+1}, 1)$, $\delta(s_{i+1}, 1) = (s_j, 1)$, and $\delta(s_j, 1) = (s_i, 1)$ (see Figure 4).

$0^{i-1}10^{K-i-1}00^{i}110^{K-i-1}0 \rightarrow$
$0^{i-1}10^{K-i-1}10^{i}110^{K-i-1}1$

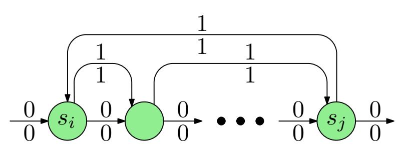


Fig. 4. The half-fixed triple segment and corresponding FST structure enforced.

The segment consists of two subsegments that each ensures a portion of the structure. The first, $0^{i-1}10^{K-i-1}0 \rightarrow 0^{i-1}10^{K-i-1}1$, and ensures that $\delta(s_i, 1) = (s_{i+1}, 1)$. The second, $0^{i}110^{K-i-1}0 \rightarrow 0^{i}110^{K-i-1}1$, ensures that $\delta(s_{i+1}, 1) = (s_j, 1)$ and $\delta(s_j, 1) = (s_i, 1)$. The state s_j cannot be in a fixed state of another half-fixed triple segment with fixed states s'_i, s'_{i+1} and free state s'_j, as then either:

- $s_j = s'_i$ and thus $\delta(s_j, 1) = (s'_{i+1}, 1) \neq (s_i, 1)$ (and thus the segment $0^{i-1}10^{K-i-1}0 \rightarrow 0^{i-1}10^{K-i-1}1$ is not transduced).
- $s_j = s'_{i+1}$ and $(s'_j, 1) = \delta(s'_{i+1}, 1) = (s_i, 1)$, so $\delta(s'_j, 1) = (s_{i+1}, 1) \neq (s'_i, 1)$ (and thus the segment $0^{i'-1}10^{K-i'-1}0 \rightarrow 0^{i'-1}10^{K-i'-1}1$ is not transduced).

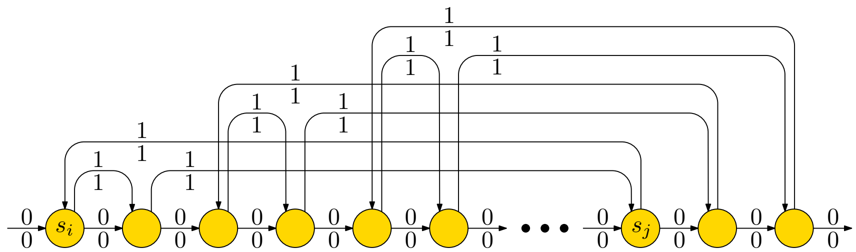


Fig. 5. The half-fixed interval segment for three consecutive free states and corresponding FST structure enforced.

Half-fixed interval. The half-fixed interval forces a collection of half-fixed triples with consecutive fixed states to also have consecutive free states. It does so by a simple traversal of the free states, checking that each has the expected pair of consecutive fixed states (see Figure 5). □

Any solution transduction for the previous reduction uses an FST where each state has at most one incoming 1-transition and 0-transition, since every transition lies on a cycle (of length 1, 3, or K). Also, any solution transduction by an FST with K states traverses $2K$ distinct transitions (with $K-1$ $(0,0)$-transitions, 1 $(0,1)$-transition, and K $(1,1)$-transitions). Any other solution FST must have at least $2K$ states and traverse at least $2K+1$ distinct transitions: $2K$ 0-transitions and at least one 1-transition. Thus the following problem is also NP-hard by the prior reduction:

*Problem 5 (*Promise encoding by FST*).* Given two strings S, S' and an integer K with the following promises about any FST T with at most K states transducing S to S', does such a T exist?

- Each state of T has at most one incoming 0-transition.
- Each state of T has at most one incoming 1-transition.
- When transducing S to S':
  - $K-1$ distinct $(0,0)$-transitions are used.
  - K distinct $(1,1)$-transitions are used.
  - 1 distinct $(0,1)$-transition is used.
  - The transitions are traversed in a unique specified order given as part of the input.

Corollary 1. *The* Promise encoding by FST *problem is* NP*-hard.*

4 Height-2 PATS is NP-complete

Göös and Orponen [5] establish that all the variations of the PATS problem considered here are in NP. So we need only consider their NP-hardness.

Theorem 2. *The non-uniform height-2* PATS *problem is* NP*-hard.*

Proof. The pattern output by the reduction consists of a bottom row encoding S and a top row encoding the sequence of transitions traversed when transducing S to S' (provided as part of the Promise encoding by FST instance). The bottom row encoding uses two colors, pink and red, corresponding to the two symbols in S. The top row encoding uses $2K$ colors, one for each transition used in the transduction of S to S'. The number of tile types permitted is $T = 2K+2$: one type per color.

The north glues of the bottom row either encode S (distinct north glues for the pink and red tile types) or $0^{|S|}$ (same glue). The latter is impossible, since then the leftmost $|S|$ locations of the top row are filled by many repetitions of the same K transitions. So the north glues of the bottom row encode S.

A set of $2K$ tile types that assemble the top row is equivalent to a set of $2K$ transitions transducing S to S', with source and destination states corresponding to west and east glues. So the top row can be assembled using $2K$ tile types exactly when S can be transduced to S' using $2K$ transitions of the specified types traversed in the specified order. Thus the pattern can be assembled using a tile set of at most $2K$ types exactly when the corresponding instance of Promise encoding by FST has a solution transducer. ◻

Theorem 3. *The uniform height-*2 PATS *problem is* NP*-hard.*

The addition of more rows with a new common color and increasing T by 1 suffices to prove both the uniform and non-uniform variants NP-hard for greater heights.

5 Uniform Height-2 3-PATS is NP-complete

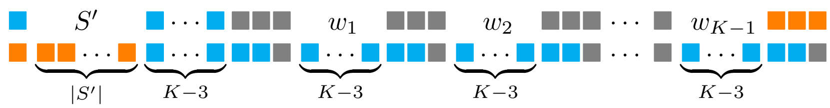
*Problem 6 (*Modified promise encoding by FST*).* Given two strings S, S' and an integer $K \not\equiv 0 \pmod 3$ with the following promises about any FST T with at most K states transducing S to S', does such a T exist?

- The first and last symbols of S' are 2.
- Each state of T has at most one incoming 0-transition.
- Each state of T has at most one incoming 1-transition.
- Every $(1,1)$-transition lies on a 1-cycle or 3-cycle of $(1,1)$-transitions.
- When transducing S to S':
  - $K-1$ distinct $(0,0)$-transitions are used.
  - $K-1$ distinct $(1,1)$-transitions are used.
  - 1 distinct $(0,1)$-transition is used.
  - 1 distinct $(1,2)$-transition is used.
  - The transitions are traversed in a unique specified order given as part of the input.

Lemma 2. *The* Modified promise encoding by FST *problem is* NP*-hard.*

Theorem 4. *The uniform height-2, 3-PATS problem is* NP*-hard.*

Proof. Let P be the following width-$(1+|S'|+K^2)$, height-2 pattern over 3 colors $\{\blacksquare, \blacksquare, \blacksquare\}$:



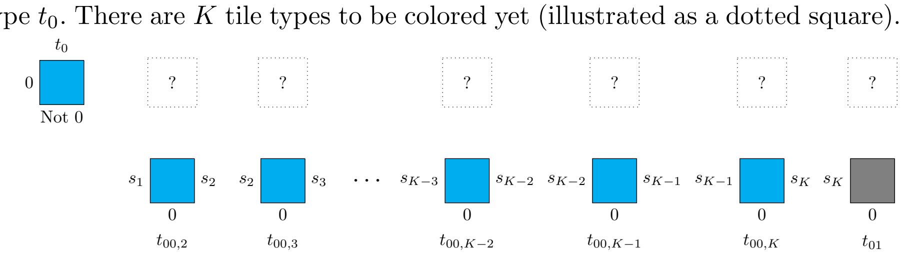
where, for $1 \le i \le K-1$,

$$w_i = \begin{cases} \blacksquare^{(3i \bmod K)-1} \blacksquare \blacksquare^{K-3-(3i \bmod K)} & \text{if } 3i \le K-3 \pmod{K} \\ \blacksquare^{K-3} & \text{otherwise.} \end{cases}$$

Notice that, for any $1 \le i < j \le K$, w_i and w_j differ in the position of 1. Split the pattern P into the leftmost $|S'|+1$ columns and the remainder, called the *transduction* and *FST-constructor* gadgets, respectively. The FST-constructor gadget is further partitioned into K rectangular subpatterns of width K.

Next, consider the constraints on RTASs with at most $|S'|+2K+2$ tile types that uniquely self-assemble P. Lemma 1 of Göös and Orponen [5] states that any smallest RTAS that uniquely self-assembles a pattern is directed. As we will prove, directed RTASs uniquely self-assembling P have size at least $|S'|+2K+2$ tile types; thus we need only consider directed systems.

Let the north and east glues of the seed be 0. The leftmost $|S'|+1$ locations in the bottom row of P are orange, with a cyan location following. So these positions must be tiled with orange tiles of pairwise-distinct type; the need for $|S'|+1$ distinct orange tile types thus arises. Similarly, the leftmost $K-1$ cyan locations in the bottom row must use $K-1$ distinct cyan tile types. These tile types share the south glue 0, and since the system is directed, their west glues are pairwise distinct. Label these $K-1$ cyan tile types left-to-right $t_{00,2}, t_{00,3}, \ldots, t_{00,K}$ and the gray tile type immediately right t_{01}, as seen below.[4] The cyan tile in the northwest corner of P cannot have the same type as any of these $K-1$ types, since otherwise this tile can also appear in the southwest corner of P. Call this type t_0. There are K tile types to be colored yet (illustrated as a dotted square).



These K tile types will turn out to be necessary, implying $(|S'|+1)+(K-1)+2+K-1+1 = S'+2K+2$ types total with $K-1$ colored gray and one colored

[4] In these later labels, the first subscript indicates the kind of transition of the FST that the tile type will be shown to simulate, e.g., $t_{00,i}$ is a $(0,0)$-transition, t_{01} and $(0,1)$-transition, etc.

orange. For this, we claim that the bottom row of all blocks but the first assemble identically by establishing that the gray tiles attaching to the southeast corner of the first two blocks are identical. Suppose not. Then the bottom row of the second block cannot reuse cyan tile types used in the bottom row of the first block. So the uncolored K tile types must be one gray and $K-1$ cyan types with south glue 0. Thus the complete tile set includes only two gray tile types with the south glue 0.

Consider the gray tile attaching at the northeast corner of the first block. Its south glue is 0 and its west glue is equal to the east glue of the gray tile attaching to its immediate left. This contradicts the directedness of the system, since a cyan tile is provided with the same pair of west and south glues. Indeed, both gray tile types appear at the southeast corner of a block and to their east are cyan tiles attaching.

The verified claim brings following properties for all but the first block:

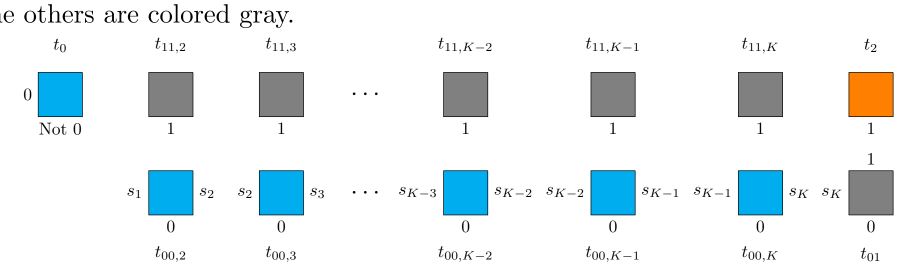
Property 1: For any $1 \leq i \leq K-1$, tiles attaching at the i-th top-row position of any two blocks but the first one have the same south glue; tiles attaching at the K-th top-row position (northeast corner) of any two blocks including the first one have the same south glue.

Property 2: Any such pair of tiles have pairwise-distinct east glues (and types).

Property 3: The assembly of the bottom row is provided with at least two different kinds of north glues.

Property 2 holds since a orange tile is placed in the northeast corner of only the last block. Thus without Property 3, $o(K^2)$ tile types would be necessary to place the orange tile. Observe that for each $1 \leq i \leq K-3$, the i-th position of exactly one block is gray and the counterpart of all other blocks are cyan; for each $K-2 \leq i \leq K$, the i-th position of only the last block is orange and the counterpart of all others are gray. Thus, Properties 1 and 2 imply that the tile type set must contain one orange and $K-1$ gray tile types whose south glue is equal to the north glue of t_{01} and one gray and $K-2$ cyan tile types with a common south glue.

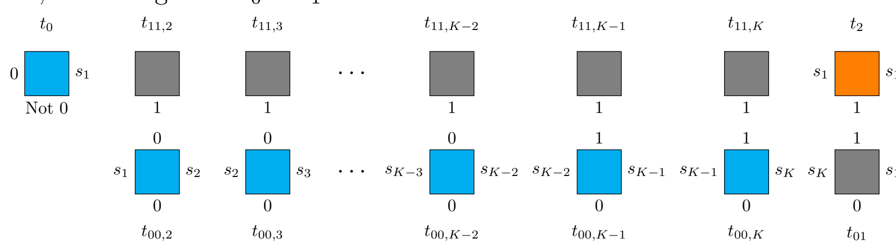
We claim these requirements enforce that the north glue of t_{01} is not 0. Suppose otherwise. Then the former requirement implies $K-2$ extra gray tile types with south glue 0. So at most 3 tile types, including t_0, have the non-0 south glue, and Property 3 cannot be satisfied. Thus, the north glue of t_{01} is not 0; call it 1. Tiles attaching at the northeast corner of the blocks must all have distinct types due to Property 2, and now also their south glues must be 1. The K uncolored tile types thus have south glue 1, and one is colored orange and all the others are colored gray.



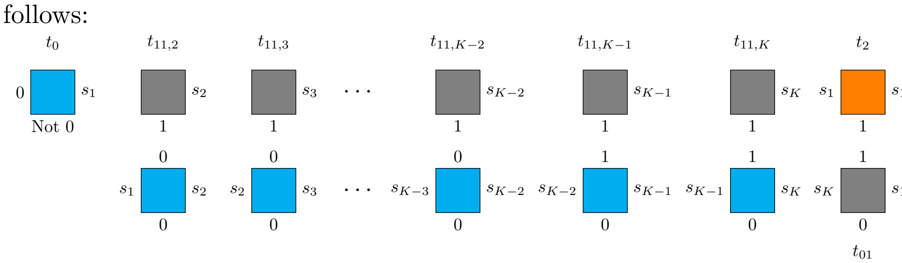
Note that a t_0 tile cannot attach anywhere in the blocks. Indeed, it causes glue mismatch with the seed being placed on the bottom row, and in order for it to attach on the top row, it must share its south glue with $K-3$ cyan tile types due to Properties 1 and 2. In summary, any minimum tile set uniquely assembling P consists of K cyan tile types, K gray ones, and $|S'|+2$ orange ones.

Now we prove constraints on the glues of these types. With only $K-1$ cyan tile types with south glue 0, even the first block must assemble its bottom row as other blocks do. That is, the bottom row of all blocks assemble as $t_{00,2}t_{00,3}\cdots t_{00,K}t_{01}$. Thus, the east glue of t_{01} is equal to the west glue of $t_{00,2}$, that is, s_1. Since t_0 does not appear in any block, Property 2 implies that the north glues of $t_{00,2}, t_{00,3}, \ldots, t_{00,K-2}$ are 0 and that the north glues of $t_{00,K-1}$ and $t_{00,K}$ are 1.

The top row of the last block is w_K■■■ $=$ ■$^{K-4}$■■■■ and its last four positions ■■■■ are assembled as $t_{01}t_2t_2t_2$. This imposes that both the east and west glues of t_2 must be equal to the east glue of t_{01}, that is, s_1. Since S' begins with 2, the east glue of t_0 is s_1.



Since S' ends with 2, no tile (necessarily of type $t_{11,2}, t_{11,3}, \ldots, t_{11,K}$ by Properties 1 and 2) appearing at the northeast corner of a block has east glue s_1. Moreover, tiles attaching to their east are of type $t_{00,2}, \ldots, t_{00,K}$ or t_{01}, thus their east glues are in $\{s_2, s_3, \ldots, s_K\}$. Without loss of generality, assign them as follows:



So the east glues of all tile types in the FST-construction gadget are in $\{s_1, \ldots, s_K\}$. The east glues of $t_{11,2}, \ldots, t_{11,K}$ are distinct and selected from $\{s_1, \ldots, s_K\}$. Since $t_{11,2}, \ldots, t_{11,K}$ share the south glue 1 with t_2, the west glue of t_2 is s_1, and the system is directed, the west glues are distinct and from $\{s_2, s_3, \ldots, s_K\}$.

The only remaining flexibility in the design of the tile set is assigning west glues to $t_{11,2}, \ldots, t_{11,K}$, corresponding to the assignment of $(1,1)$-transition sources in Modified promise encoding by FST. All that remains is to prove this correspondence indeed holds.

The glue 0 is not in $\{s_1, \dots, s_K\}$, as otherwise a cyan or gray tile could appear in the southwest corner of P. So none of the cyan, gray, or t_2 tile types has east glue 0 and thus a t_0 tile cannot attach anywhere but the northwest corner of P. Also, observe that the tile set constraints imply that the north glue of a tile in the bottom row can be discerned by examining the color of the tile north and (possibly) northeast (a cyan tile north implies a 0 glue, a gray tile north implies a 0 or 1 glue if the color of the northeast tile is cyan or orange, respectively).

Finally, consider the tile types, excluding t_0 and all orange tile types except t_2, as a set of transitions of an FST, with $t_{00,i}$ types as $(0,0)$-transitions, $t_{11,i}$ types as $(1,1)$-transitions, t_{01} a $(0,1)$-transition, and t_2 a $(1,2)$-transition. The constraints induced on the resulting transitions (e.g. that there are $K-1$ $(1,1)$-transitions because there are $K-1$ color-1 tile types with south glue 1) is found as a constraint on the transitions in the statement of Lemma 2. In particular, the choice of w_i's in consecutive blocks requires that every $(1,1)$-transition lies on a 1-cycle or 3-cycle of $(1,1)$-transitions (the last constraint of Lemma 2). Thus there exists a solution FST to the Modified promise encoding by FST instance if and only if there exists a solution tile set. □

6 Efficiently Solvable PATS Problems

The non-uniform height-1 PATS problem is trivially solvable using one tile type for each color. This idea can be generalized for all patterns of fixed height:

Theorem 5. *The non-uniform height-h c-PATS problem can be solved in $c^{c^{O(h)}}n$ time.*

As established in Section 5, a similar algorithm for the uniform model is impossible unless P = NP. Nevertheless, the uniform height-1 PATS problem can be solved in linear time using a pigeonhole argument and a DFS-based search for the longest repetitive suffix of a given height-1 pattern:

Theorem 6. *The uniform height-1 PATS problem can be solved in $O(n)$ time.*

7 Conclusion

Our work here extends the extensive prior work on the parameterized c-PATS problem to also incorporate pattern height and uniformity, and finds a more delicate complexity landscape: limited height and colors do not make the PATS problem tractable, except when combined in the non-uniform model, or in degenerate cases (height-1 or 1-PATS). A single combination of parameters and model remains unresolved; we conjecture the following:

Conjecture 1. The uniform height-2 2-PATS problem is NP-hard.

We encourage further parameterized analysis of problems in tile self-assembly in support of recent efforts in developing a more complete understanding of the structural complexity of tile self-assembly (see [14]).

Acknowledgments

We thank Yo-Sub Han for very fruitful discussions about finite automata and tile self-assembly, and anonymous reviewers for comments that improved the paper.

References

1. D. Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 39:337–350, 1978.
2. E. Czeizler and A. Popa. Synthesizing minimal tile sets for complex patterns in the framework of patterned DNA self-assembly. *Theoretical Computer Science*, 499:23–37, 2013.
3. D. Doty. Theory of algorithmic self-assembly. *Communications of the ACM*, 55(12):78–88, 2012.
4. M. R. Garey and D. S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing*, 4(4):397–411, 1975.
5. M. Göös and P. Orponen. Synthesizing minimal tile sets for patterened dna self-assembly. In *Proceedings of the 16th International Conference on DNA Computing and Molecular Programming (DNA 16)*, volume 6518 of *LNCS*, pages 71–82. Springer, 2010.
6. A. C. Johnsen, M.-Y. Kao, and S. Seki. Computing minimum tile sets to self-assemble color patterns. In *Proceedings of the 24th International Symposium on Algorithms and Computation (ISAAC)*, volume 8283 of *LNCS*, pages 699–710. Springer, 2013.
7. A. C. Johnsen, M.-Y. Kao, and S. Seki. A manually-checkable proof for the NP-hardness of 11-color pattern self-assembly tile set synthesis. *Journal of Combinatorial Optimization*, 2015. In press.
8. L. Kari, S. Kopecki, and S. Seki. Binary pattern tile set synthesis is NP-hard. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 9134 of *LNCS*, pages 1022–1034. Springer, 2015.
9. X. Ma and F. Lombardi. Synthesis of tile sets for dna self-assembly. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(5):963–967, 2008.
10. M. J. Patitz. An introduction to tile-based self-assembly. In *Proceedings of Unconventional Computation and Natural Computation (UCNC) 2012*, volume 7445 of *LNCS*, pages 34–62. Springer, 2012.
11. S. Seki. Combinatorial optimization in pattern assembly. In *Proceedings of Unconventional Computation and Natural Computation (UCNC) 2013*, volume 7956 of *LNCS*, pages 220–231. Springer, 2013.
12. U. V. Vazirani and V. V. Vazirani. A natural encoding scheme proved probabilistic polynomial complete. *Theoretical Computer Science*, 24(3):291–300, 1983.
13. E. Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, Caltech, 1998.
14. D. Woods. Intrinsic universality and the computational power of self-assembly. *Philosophical Transaction of the Royal Society A*, 373(2046), 2015.