

Two Hands Are Better Than One (up to constant factors): Self-Assembly In The 2HAM vs. aTAM

Sarah Cannon¹, Erik D. Demaine², Martin L. Demaine³, Sarah Eisenstat⁴, Matthew J. Patitz⁵, Robert T. Schweller⁶, Scott M. Summers⁷, and Andrew Winslow⁸

- 1 Mathematical Institute, University of Oxford, 24-29 St. Giles', Oxford OX1 3LB, UK sarah.cannon@linacre.ox.ac.uk. Research supported in part by National Science Foundation grants CCF-0830734 and CBET-0941538.
- 8 Department of Computer Science, Tufts University, Medford, MA 02155, USA, {scanno01,awinslow}@cs.tufts.edu. Research supported in part by National Science Foundation grants CCF-0830734 and CBET-0941538.
- 2,3,4Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar St., Cambridge, MA 02139, USA, {edemaine,mdemaine,seisenst}@mit.edu. Research supported in part by NSF grant CDI-0941538.
- 5 Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701, USA, patitz@uark.edu. Research supported in part by National Science Foundation grant CCF-1117672.
- 6 Department of Computer Science, University of Texas–Pan American, Edinburg, TX, 78539, USA, rtschweller@utpa.edu. Research supported in part by National Science Foundation grant CCF-1117672.
- 7 Department of Computer Science and Software Engineering, University of Wisconsin–Platteville, Platteville, WI 53818, USA, summerss@uwplatt.edu

Abstract

We study the difference between the standard seeded model (aTAM) of tile self-assembly, and the “seedless” two-handed model of tile self-assembly (2HAM). Most of our results suggest that the two-handed model is more powerful. In particular, we show how to simulate any seeded system with a two-handed system that is essentially just a constant factor larger. We exhibit finite shapes with a busy-beaver separation in the number of distinct tiles required by seeded versus two-handed, and exhibit an infinite shape that can be constructed two-handed but not seeded. Finally, we show that verifying whether a given system uniquely assembles a desired supertile is co-NP-complete in the two-handed model, while it was known to be polynomially solvable in the seeded model.

1998 ACM Subject Classification F.1.2

Keywords and phrases abstract tile assembly model, hierarchical tile assembly model, two-handed tile assembly model, algorithmic self-assembly, DNA computing, biocomputing, Wang tiles

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Algorithmic self-assembly is a burgeoning area that studies how to computationally design geometric systems of simple parts that self-assemble into desired complex shapes or functionalities. The field began with Erik Winfree’s PhD thesis [17] and two STOC papers

about a decade ago [2, 15]. The theoretical models introduced in this work have since been implemented in real molecular systems using DNA tiles [5, 16]. From a practical perspective, these systems are exciting because they enable controlled manufacture of precise geometric objects at nanometer resolution (*nanomanufacture*). From a theoretical Computer Science perspective, this area is exciting because it offers a model of computation where the computer consists of geometric objects, which is challenging to work with because the allowed operations are highly constrained (simple, local interactions between the objects), yet there are many results classifying the difficulty of assembling many different shapes.

1.1 A tale of two models.

Most work in algorithmic self-assembly uses the *abstract Tile Assembly Model (aTAM)* [2, 15, 17]. In this model, the core of a self-assembly system is a set of *Wang tiles*—unit squares with up to one *glue* (label) on each edge, with each type available in infinite supply. One such tile is marked as a *seed* (starting point) of a single assembly, and the model defines how tiles can repeatedly attach to this assembly (according to glue strengths and an overall temperature—see Section 2.1 for details), which ultimately becomes the (usually single) output of the system.

In reality, tiles mix in solution according to Brownian motion, and attractive forces cause them to fuse into larger assemblies. Presumably, the aTAM defines a seed tile to keep track of a single assembly instead of the many copies assembled in reality (as seen in the atomic force microscopy images in [5, 16]). However, as a side effect, the aTAM fails to capture the possibility that multiple assemblies grow (e.g., from multiple copies of the seed) and attach to each other, potentially making unintended assemblies not predicted by the aTAM. In addition, the ability to fuse larger assemblies in reality could potentially be exploited to design more efficient self-assembly systems for a desired shape. These possible discrepancies between the aTAM and reality are the topic of this paper.

The *Two-Handed Tile Assembly Model (2HAM)* [1, 7, 9, 10, 12, 13] (also known as *Hierarchical Self-Assembly* [6]) is essentially an unseeded generalization of the aTAM, in which any two assemblies (including but not limited to individual tiles) can fuse to each other. Instead of using seeds, the 2HAM defines the “output” of the system to consist of all assemblies that cannot be fused with any others possibly produced by the system. (See Section 2.2 for the definition.) This model captures the possibility of larger assemblies fusing together, although it remains to be studied whether it accurately models reality.¹

1.2 Our results.

The central problem addressed in this paper is to determine the difference in theoretical power between these two models of self-assembly: the aTAM and the 2HAM. In particular we show that, up to constant factors, many results in the standard aTAM can be converted to apply in the 2HAM. On the other hand, we show that the 2HAM enables substantially more efficient self-assembly systems in some cases than what is possible in the aTAM. We conclude that two hands are better than one, up to constant factors.

Our main results are the following (see Tables 1, 2, and 3 for additional results):

¹ 2HAM does not model the “floppiness” of assemblies (i.e. non-rigidity), which may allow bending that prevents proper alignment of glues or shifting of potentially blocking portions between two larger assemblies. It also ignores the reduced speed and/or concentration of larger assemblies, which may substantially impact the time required for assembly.

aTAM systems	Simulating 2HAM systems
$\tau \in \{1, 2\}$	$\tau = 2$, scale factor 5(thm. 4.2)
$\tau = 3$	$\tau = 3$, scale factor 5(thm. 4.3)
$\tau \geq 4$	$\tau = 4$, scale factor 5(thm. 4.1)

■ **Table 1** Summary of results for simulating the aTAM model using the 2HAM model.

	Loops		Staircases	Infinite staircases	
	$\tau = 1$	$\tau = 2$	$\tau = 2$	Finite assembly	Self-assembly
aTAM	$n + 5$	$n + 3$	2^n steps: $\Omega\left(\frac{n}{\log n}\right)$ (thm. 3.2)	No (thm. 3.6)	No (thm. 3.6)
2HAM	$2n + 2$	$\leq n + 3$	$2^{O(\text{running time of } M \text{ on } x)}$ steps: $O(Q + x)$ (thm. 3.3)	Yes ($\tau = 2$) (thm. 3.5)	Open

■ **Table 2** Summary of results showing separation between the aTAM and 2HAM with respect to tile complexity. The value of a cell denotes the tile complexity. Note that some of our results are asymptotic while others are exact complexities. The term *Finite assembly* refers to *finite self-assembly*, which is defined in Section 2. For infinite staircases, “Yes” means the structure does self-assemble, “No” means that it does not self-assemble and “Open” means the question is open. Note that, for table cells that do not contain a reference, the theorem and corresponding proof are omitted from this version of the paper due to space constraints.

Simulation: [Section 4, Table 1]

1. Any aTAM system with temperature $\tau \geq 2$ can be simulated by a 2HAM system with the same temperature τ , which produces a 5×5 scaled version of the same shape plus a portion of a unit-thickness “coating”.
2. Any aTAM system with temperature $\tau \geq 4$ can be simulated by a 2HAM system with a temperature of 4. Thus low-temperature 2HAM is at least as powerful as even high-temperature aTAM, up to constant-factor scale.

Separation: [Section 3, Table 2]

3. There is a shape that can be assembled in the aTAM at temperature $\tau = 1$ using $n + 5$ unique tile types but any 2HAM system in which the shape assembles at the same temperature requires $2n + 2$ unique tile types. At temperature $\tau = 2$, the same shape can be assembled in both models using $n + 3$ tile types.
4. There is a shape that can be assembled in the 2HAM using n tile types, while the number of tile types required for any aTAM assembly of the shape is (roughly) exponential in n . This result can be extended to show that there is a shape that can be built in the 2HAM using $O(n)$ tile types, but in the aTAM the same shape requires $BB(n)$ tile types, where $BB(n)$ is the busy beaver function.
5. There is an infinite shape that can self-assemble in the aTAM but not in the 2HAM. Note that this does not contradict our first simulation result because our simulation scales up the simulated system by a constant factor.
6. There is an infinite shape that can self-assemble (in a weaker sense) in the 2HAM but not in the aTAM.

Verification: [Section 5, Table 3]

7. It is co-NP-complete to determine whether a given 2HAM self-assembly system uniquely assembles a given 3D supertile (the Unique Assembly problem is co-NP-complete in the 2HAM), while the same problem is known to be polynomial time solvable for aTAM [3]²

² Adleman et. al. [3] actually considered a slight variant of the Unique Assembly problem in which the

	Producible		Unique Assembly		Unique Shape	
	$\tau = 1$	$\tau = 2$	$\tau = 1$	$\tau = 2$	$\tau = 1$	$\tau = 2$
aTAM	$O(a)$		$O(a^2 + at)$ [3]		co-NPC	co-NPC [7]
2HAM	$O(at)$	$O(a^4)$ [11]	$O(ta^2 + at^2)$	co-NPC (thm. 5.1)	co-NP [7]	co-NPC [7]
	Terminal		Finite Existence		Infinite Existence	
	$\tau = 1$	$\tau = 2$	$\tau = 1$	$\tau = 2$	$\tau = 1$	$\tau = 2$
aTAM	$O(at)$ [3]		UC		UC	
2HAM	$O(at)$	UC	UC		Open	UC

■ **Table 3** Complexities of assembly verification problems for the aTAM and 2HAM. The variable a denotes the size of an input assembly, and τ and t denote the temperature and tileset size for an input aTAM or 2HAM system. “UC” stands for uncomputable. Note that, for table cells that do not contain a reference, the theorem and corresponding proof are omitted from this version of the paper due to space constraints.

(This result is the only one in 3D; all other results are in 2D.) We provide results for the complexity for five additional verification problems for the aTAM and the 2HAM.

This paper aims to be a first major step toward a thorough “complexity theory” for self-assembly. Like traditional complexity theory, there are several potential models for self-assembly, and we need to understand the relative power among these models. Even our definition of “simulation” is new in that it is the first to also handle the dynamics of systems such as the 2HAM, and we hope that it forms the foundation for further such results.

2 Preliminaries and notation

We work in the 2-dimensional discrete space \mathbb{Z}^2 . Define the set $U_2 = \{(0, 1), (1, 0), (0, -1), (-1, 0)\}$ to be the set of all *unit vectors* in \mathbb{Z}^2 . We also sometimes refer to these vectors by their cardinal directions N, E, S, W , respectively. All *graphs* in this paper are undirected. A *grid graph* is a graph $G = (V, E)$ in which $V \subseteq \mathbb{Z}^2$ and every edge $\{\vec{a}, \vec{b}\} \in E$ has the property that $\vec{a} - \vec{b} \in U_2$.

Intuitively, a tile type t is a unit square that can be translated, but not rotated, having a well-defined “side \vec{u} ” for each $\vec{u} \in U_2$. Each side \vec{u} of t has a “glue” with “label” $\text{label}_t(\vec{u})$ —a string over some fixed alphabet—and “strength” $\text{str}_t(\vec{u})$ —a nonnegative integer—specified by its type t . Two tiles t and t' that are placed at the points \vec{a} and $\vec{a} + \vec{u}$ respectively, *bind* with *strength* $\text{str}_t(\vec{u})$ if and only if $(\text{label}_t(\vec{u}), \text{str}_t(\vec{u})) = (\text{label}_{t'}(-\vec{u}), \text{str}_{t'}(-\vec{u}))$.

In the subsequent definitions, given two partial functions f, g , we write $f(x) = g(x)$ if f and g are both defined and equal on x , or if f and g are both undefined on x .

Fix a finite set T of tile types. A T -*assembly*, sometimes denoted simply as an *assembly* when T is clear from the context, is a partial function $\alpha : \mathbb{Z}^2 \dashrightarrow T$ defined on at least one input, with points $\vec{x} \in \mathbb{Z}^2$ at which $\alpha(\vec{x})$ is undefined interpreted to be empty space, so that $\text{dom } \alpha$ is the set of points with tiles. We write $|\alpha|$ to denote $|\text{dom } \alpha|$, and we say α is *finite* if $|\alpha|$ is finite. For assemblies α and α' , we say that α is a *subassembly* of α' , and write $\alpha \sqsubseteq \alpha'$, if $\text{dom } \alpha \subseteq \text{dom } \alpha'$ and $\alpha(\vec{x}) = \alpha'(\vec{x})$ for all $x \in \text{dom } \alpha$.

For $\tau \in \mathbb{N}$, an assembly is τ -*stable* if every cut of its binding graph has strength at least

input is a shape and the output is whether or not the input system uniquely assembles one supertile with that shape. Within the aTAM, the complexity of this variant problem is polynomially related to our problem. In contrast, this is not clearly the case in the 2HAM, making this variant problem a potentially interesting direction for future work. Further, [3] call their problem the Unique Shape problem, which is not the same as our version of the Unique Shape problem in that we do not require the input system be directed. Our version of the Unique Shape problem was first considered in [7].

τ , where the weight of an edge is the strength of the glue it represents. That is, the supertile is stable if at least energy τ is required to separate the supertile into two parts.

2.1 Informal description of the abstract tile assembly model (aTAM)

In this section we give an informal description of the aTAM. The reader is encouraged to see [14, 15, 17] for a formal development of the model.

In the aTAM, self-assembly begins with a *seed assembly* σ (typically assumed to be finite and τ -stable) and proceeds asynchronously and nondeterministically, with tiles adsorbing one at a time to the existing assembly in any manner that preserves stability at all times.

An aTAM *tile assembly system* (TAS) is an ordered triple $\mathcal{T} = (T, \sigma, \tau)$, where T is a finite set of tile types, σ is a seed assembly with finite domain, and τ is the temperature. An *assembly sequence* in a TAS $\mathcal{T} = (T, \sigma, \tau)$ is a (possibly infinite) sequence $\vec{\alpha} = (\alpha_i \mid 0 \leq i < k)$ of assemblies in which $\alpha_0 = \sigma$ and each α_{i+1} is obtained from α_i by the “ τ -stable” addition of a single tile. The *result* of an assembly sequence $\vec{\alpha}$ is the unique assembly $\text{res}(\vec{\alpha})$ satisfying $\text{dom } \text{res}(\vec{\alpha}) = \bigcup_{0 \leq i < k} \text{dom } \alpha_i$ and, for each $0 \leq i < k$, $\alpha_i \sqsubseteq \text{res}(\vec{\alpha})$.

We write $\mathcal{A}[\mathcal{T}]$ for the *set of all producible assemblies of \mathcal{T}* . An assembly α is *terminal*, and we write $\alpha \in \mathcal{A}_{\square}[\mathcal{T}]$, if no tile can be stably added to it. We write $\mathcal{A}_{\square}[\mathcal{T}]$ for the *set of all terminal assemblies of \mathcal{T}* . A TAS \mathcal{T} is *directed*, or *produces a unique assembly*, if it has exactly one terminal assembly i.e., $|\mathcal{A}_{\square}[\mathcal{T}]| = 1$. The reader is cautioned that the term “directed” has also been used for a different, more specialized notion in self-assembly [4]. We interpret “directed” to mean “deterministic”, though there are multiple senses in which a TAS may be deterministic or nondeterministic.

Given a connected shape $X \subseteq \mathbb{Z}^2$, we say a TAS \mathcal{T} *self-assembles* X if every producible, terminal assembly places tiles exactly on those positions in X . (Note that this notion is equivalent to *strict* self-assembly as defined in [14].) For an infinite shape $X \subseteq \mathbb{Z}^2$, we say that \mathcal{T} *finitely self-assembles* X if every finite producible assembly of \mathcal{T} has a possible way of growing into an assembly that places tiles exactly on those points in X . Note that if a shape X self-assembles in \mathcal{T} , then X finitely self-assembles in \mathcal{T} .

2.2 Informal description of two-handed tile assembly model (2HAM)

The 2HAM [1, 7, 9, 10, 12, 13] is a generalization of the aTAM in that it allows for two assemblies, both possibly consisting of more than one tile, to attach to each other. Since we must allow that the assemblies might require translation before they can bind, we define a *supertile* to be the set of all translations of a τ -stable assembly, and speak of the attachment of supertiles to each other, modeling that the assemblies attach, if possible, after appropriate translation. We now give a brief, informal, sketch of the 2HAM.

A *supertile* (a.k.a., *assembly*) is a positioning of tiles on the integer lattice \mathbb{Z}^2 . Two adjacent tiles in a supertile *interact* if the glues on their abutting sides are equal and have positive strength. Each supertile induces a *binding graph*, a grid graph whose vertices are tiles, with an edge between two tiles if they interact. The supertile is *τ -stable* if it is τ -stable in the sense of aTAM. A 2HAM *tile assembly system* (TAS) is a pair $\mathcal{T} = (T, \tau)$, where T is a finite tile set and τ is the *temperature*, usually 1 or 2. Given a TAS $\mathcal{T} = (T, \tau)$, a supertile is *producible*, written as $\alpha \in \mathcal{A}[\mathcal{T}]$ if either it is a single tile from T , or it is the τ -stable result of translating two producible assemblies without overlap.³ A supertile α is *terminal*, written

³ The restriction on overlap is our formalization of the physical mechanism of steric protection.

as $\alpha \in \mathcal{A}_\square[\mathcal{T}]$ if for every producible supertile β , α and β cannot be τ -stably attached. A TAS is *directed* if it has only one terminal, producible supertile.

Given a connected shape $X \subseteq \mathbb{Z}^2$, we say a TAS \mathcal{T} *self-assembles* X if it self-assembles in the sense of aTAM (appropriately translated if necessary). For an infinite shape $X \subseteq \mathbb{Z}^2$, we say that \mathcal{T} *finitely self-assembles* X if it finitely self-assembles in the sense of aTAM (appropriately translated if necessary).

3 Are two hands more (tile) efficient than one?

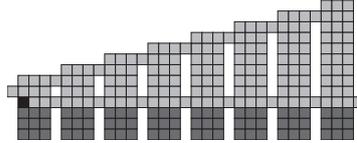
From a theoretical perspective, is the 2HAM “better” than the aTAM in terms of the minimum number of tiles required to uniquely produce a target shape? Is it possible to build certain infinite shapes in one model but not the other? Or perhaps is it possible to build finite shapes more (tile) efficiently in one model than the other? These are the questions that motivate this section.

We find, somewhat surprisingly, that it is possible for both models to “win”, in the sense that there exist shapes that self-assemble more efficiently in the aTAM than the 2HAM, and vice versa, depending on both the choice of shape as well as temperature value. At temperature $\tau = 1$, we discover an $O(1)$ separation between the aTAM and 2HAM in favor of the aTAM winning. At temperature $\tau > 1$, we see a nearly exponential (and beyond) separation in favor of the 2HAM.

3.1 Finite Shapes: staircases

We first examine classes of finite shapes that “separate” the aTAM and the 2HAM with respect to the tile complexities of the systems that uniquely produce them.

Given a shape $X \subseteq \mathbb{Z}^2$, we say that $\mathcal{C}_{\text{aTAM}}^\tau(X)$ is the *tile complexity* of X in the aTAM at temperature $\tau \in \mathbb{N}$. In other words, $\mathcal{C}_{\text{aTAM}}^\tau(X) = \min\{|T| \mid \text{for some } \sigma \text{ where } |\sigma| = 1 \text{ and } X \text{ self-assembles in } \mathcal{T} = (T, \sigma, \tau)\}$. Intuitively, $\mathcal{C}_{\text{aTAM}}^\tau(X)$ is the size of the smallest tile set that produces assemblies that place tiles on—and only on—the target shape X . Let $\mathcal{C}_{\text{aTAM}}(X) = \min\{\mathcal{C}_{\text{aTAM}}^\tau(X) \mid \tau \in \mathbb{N}\}$. The quantities $\mathcal{C}_{\text{2HAM}}^\tau(X)$ and $\mathcal{C}_{\text{2HAM}}(X)$ are defined similarly.



■ **Figure 1** A staircase with 2^3 steps with each step of width 3. The black square represents the point $(0, 0)$.

► **Definition 3.1.** For each $i, k \in \mathbb{N}$, let $B_{i,k} = (\{0, \dots, k-1\} \times \{-k, \dots, 0, \dots, i+2\}) \cup \{(-1, i+1), (k, 0)\}$ and define, $S_n = \bigcup_{i=0}^{2^n-1} (B_{i,n} + ((n+1)i, 0))$. Intuitively, the set S_n is a “staircase with 2^n steps with each step of width n .” See Figure 1 for an example of S_3 .

We will use S_n to show a non-trivial (nearly) exponential separation between the aTAM and the 2HAM.

► **Theorem 3.2.** For all $n \in \mathbb{N}$, $\mathcal{C}_{\text{aTAM}}(S_n) = \Omega\left(\frac{n}{\log n}\right)$ and $\mathcal{C}_{\text{2HAM}}^2(S_n) = O\left(\frac{\log n}{\log \log n}\right)$.

We use a counting argument to prove $\mathcal{C}_{\text{aTAM}}(S_n) = \Omega\left(\frac{n}{\log n}\right)$. It is interesting to note that, if one were to apply the standard, perhaps most obvious information-theoretic argument

to prove the bound, one would only obtain a bound of $\Omega\left(\frac{\log n}{\log \log n}\right)$, which would not give more than a $O(1)$ separation between the aTAM and the 2HAM.

We get $C_{2\text{HAM}}^2(S_n) = O\left(\frac{\log n}{\log \log n}\right)$ because, in 2HAM, we can enforce pairs of connector-column tiles to attach simultaneously, which is not possible in aTAM constructions. Intuitively, the construction works as follows. We begin by using a modified version of the optimal square construction [3] to form the lower $n \times n$ square portion of each stair step. We modify the optimal square construction to allow tiles to nondeterministically attach to the top row of the square to form a length n binary string. Then we use a binary counter [2, 7] to count from the nondeterministically chosen value, say x , up to $2^{n+1} - 1$. Finally, consecutive stair steps come together, in a purely two-handed fashion, via two strength-1 glues that are separated by a distance proportional to the height of the stair step on which they are present.

We can “iterate” the basic staircase construction using Turing machines to build each stair step. This gives an even greater separation.

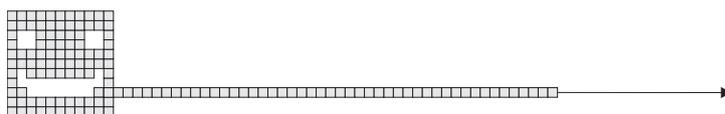
► **Theorem 3.3** (“Busy Beaver” staircase). *Let $M = (Q, \{0, 1\}, 0, \{0, 1\}, \delta, q_0, F)$ be a Turing machine and $x \in \{0, 1\}^*$ such that M halts on x . Then $C_{2\text{HAM}}^2(S_{2t(x)+|x|+2}) = O(|Q| + |x|)$, where $t(x)$ denotes the running time of M on input x .*

Theorem 3.3 says that, at temperature $\tau = 2$, the 2HAM can be used to build certain shapes much (much much...) more efficiently than in the aTAM, which requires some number of tile types nearly exponential in the number of time steps of a busy beaver Turing machine!

3.2 Infinite Shapes

In this subsection, we examine a class of infinite (staircase-like) shapes that finitely self-assemble in 2HAM but do not self-assemble in aTAM.

We first note that it is easy to exhibit a class of infinite shapes that self-assemble in the aTAM but do not self-assemble in the 2HAM. Simply take any finite shape $X \subset \mathbb{Z}^2$ and union it with a one-way infinite line to get a kind of “blob with an infinite tail” (See Figure 2 for an example of such a shape). Such shapes do not self-assemble in the 2HAM via a straightforward pumping lemma argument on the infinite tail portion of the shape. However, we note that it is easy to take any such blob+tail shape and exhibit an aTAM TAS

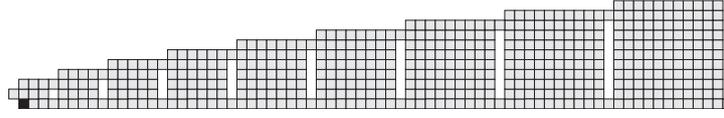


■ **Figure 2** A blob with an infinite tail.

in which that shape self-assembles. To see this, simply create hard-coded tile types for the finite blob portion (with the seed tile placed at some location in the blob) and then have a single tile type that repeats infinitely in one direction for the tail portion. This construction also testifies to the finite self-assembly of a blob+tail shape in the 2HAM.

► **Definition 3.4.** For each $i \in \mathbb{N}$, let $B_i = (\{0, \dots, i+2\} \times \{0, \dots, i+2\}) \cup \{(-1, i+1), (i, 0)\}$ and $S_\infty = \bigcup_{i=0}^{\infty} \left(B_i + \left(\left(\frac{i(i+7)}{2}, 0 \right) \right) \right)$. Intuitively, the set S_∞ is essentially a succession of larger and larger squares that are connected by pairs of tiles positioned at the top right and bottom right of each square. See Figure 3 for an example.

► **Theorem 3.5.** *The infinite staircase S_∞ finitely self-assembles in the 2HAM.*



■ **Figure 3** A finite portion of the infinite staircase, denoted as S_∞ . The black square represents the origin.

Intuitively, our construction for Theorem 3.5 proceeds as follows. We first assemble horizontal lines using three tile types: one to start the line, one to keep it going and one to stop the line. The tile that stops the line may attach non-deterministically at any step, whence lines of every length are able to form. Each line of length k ultimately grows into a $k \times k$ square. Connector-tiles that attach to the left and right of each square ensure that only a $(k - 1) \times (k - 1)$ square may attach to the left of a $k \times k$ square.

► **Theorem 3.6.** *The infinite staircase S_∞ does not finitely self-assemble in the aTAM.*

Intuitively, the proof for Theorem 3.6 is the “infinite” version of Theorem 3.2 in which there are infinitely many identical and cooperating pairs of connector tiles, and we can use really “tall” cooperating connector-columns to force “shorter” versions of identical connector-columns to grow outside of S_∞ .

► **Corollary 3.7.** *The infinite staircase S_∞ does not self-assemble in the aTAM.*

4 Simulating aTAM with 2HAM

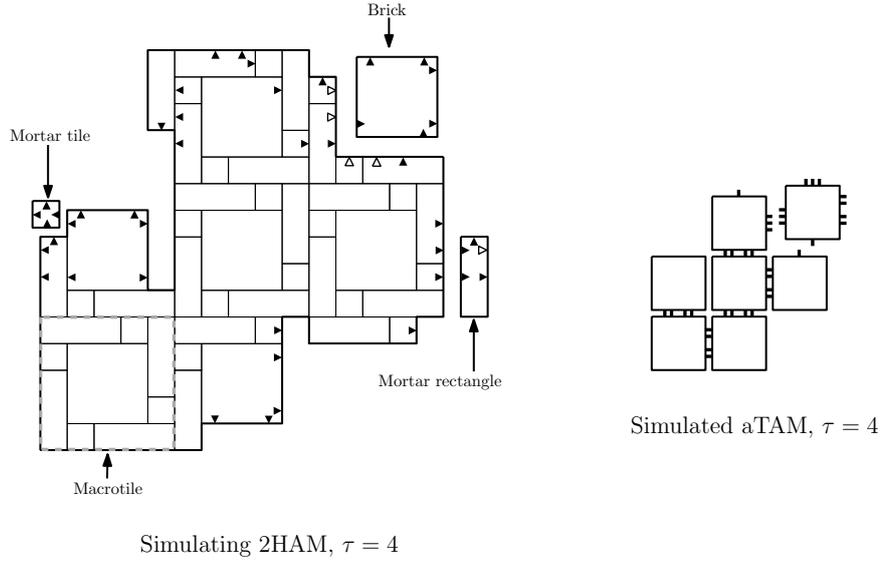
This section describes how to simulate an aTAM system by a 2HAM system, which suggests that anything the aTAM can do, the 2HAM can do (at least as good as, if not) better. A key property of our constructions is that they not only simulate the produced shapes assembled by the aTAM system, but also simulate the incremental assembly process, where single tiles aggregate on a larger seed assembly.

4.1 Simulation definition: simulate an aTAM (or 2HAM) system with another 2HAM (or aTAM) system

In this subsection, we formally define what it means for one 2HAM TAS to “simulate” another 2HAM (or aTAM) TAS. For a tileset T , let A^T and \tilde{A}^T denote the set of all assemblies over T and all supertiles over T respectively.

An m -block assembly over tile set S is a partial function $\gamma : \mathbb{Z}_m^2 \dashrightarrow S$. Let B_m^S be the set of all m -block assemblies over S . The m -block with no domain is said to be *empty*. For a general assembly $\alpha \in A^S$ define $\alpha_{x,y}^m$ to be the m -block defined by $\alpha_{x,y}^m(i, j) = \alpha(mx + i, my + j)$ for $0 \leq i, j < m$. For a partial function $R : B_m^S \dashrightarrow T$, define the *assembly replacement function* $R^* : A^S \rightarrow A^T$ such that $R^*(\alpha) = \beta$ if and only if $\beta(x, y) = R(\alpha_{x,y}^m)$ for all $x, y \in \mathbb{Z}^2$. Further, α is said to map *cleanly* to β under R^* if for all non empty blocks $\alpha_{x,y}^m$, either 1) $(x + u, y + v) \in \text{dom } \beta$ for some $u, v \in \{-1, 0, 1\}$, or 2) α has at most one non-empty m -block $\alpha_{x,y}^m$. For a given *assembly replacement function* R^* , define the *supertile replacement function* $\tilde{R} : \tilde{A}^S \rightarrow \mathcal{P}(A^T)$ such that $\tilde{R}(\tilde{\alpha}) = \{R^*(\alpha) \mid \alpha \in \tilde{\alpha}\}$. $\tilde{\alpha}$ is said to *map cleanly* to $\tilde{R}(\tilde{\alpha})$ if $\tilde{R}(\tilde{\alpha}) \in \tilde{A}^T$ and α maps cleanly to $R^*(\alpha)$ for all $\alpha \in \tilde{\alpha}$.

Consider an aTAM or 2HAM system \mathcal{S} with tileset S , and an aTAM or 2HAM system \mathcal{T} with tile set T . \mathcal{S} *simulates* \mathcal{T} at scale factor m if there exists an m -block replacement $R : B_m^S \rightarrow T$ satisfying the following conditions.



■ **Figure 4** The simulation of an assembly in an aTAM system simulated using a 2HAM system. The filled and unfilled arrows represent glues of strength 2 and 1 respectively in the 2HAM system, while the dashes each represent a bond of strength 1 in the aTAM system (i.e. 4 dashes on the North side of a tile is a glue of strength 4).

1. Equivalent Production: (1) $\{\tilde{R}(\alpha) | \alpha \in \mathcal{A}[\mathcal{S}]\} = \mathcal{A}[\mathcal{T}]$ and (2) for all $\alpha \in \mathcal{A}[\mathcal{S}]$, α maps cleanly to $\tilde{R}(\alpha)$
2. Equivalent Dynamics: (1) For any $\alpha, \alpha' \in \mathcal{A}[\mathcal{S}]$ such that $\alpha \rightarrow_{\mathcal{S}}^1 \alpha'$, then $\tilde{R}(\alpha) \rightarrow_{\mathcal{T}}^{\leq 1} \tilde{R}(\alpha')$, and (2) for any $\beta, \beta' \in \mathcal{A}[\mathcal{T}]$ such that $\beta \rightarrow_{\mathcal{T}}^1 \beta'$, then for all α such that $\tilde{R}(\alpha) = \beta$, there exists an α'' such that $\tilde{R}(\alpha'') = \beta$, $\alpha'' \rightarrow_{\mathcal{S}} \alpha'$, and $\alpha'' \rightarrow_{\mathcal{S}}^1 \alpha'$ for some α' with $\tilde{R}(\alpha') = \beta'$.

4.2 Simulating aTAM at $\tau \geq 4$ with 2HAM $\tau = 4$

It is possible to simulate the aTAM at temperature $\tau \geq 4$ using the 2HAM at temperature 4 with a constant scale factor of 5. Given any aTAM system, each tile t in the aTAM system is represented by 25 tiles forming a 5×5 macrotile assembly in the 2HAM system. The macrotile in the 2HAM system consists of a 3×3 center *brick* assembly, surrounded on all sides by a *mortar* one tile thick. These tiles are designed such that bricks and certain mortar pieces can assemble independently, but bricks cannot attach to mortar pieces or other bricks unless additional tiles are present.

We mimic the seeded nature of aTAM systems by allowing the mortar to assemble around a seed brick corresponding to the seed tile in the aTAM system by strengthening the glues at this seed macrotile. Once any brick has its complete set of mortar pieces attached to it, mortar pieces for adjacent tiles can attach to the assembly; new bricks can then attach to this partially built assembly only once their mortar is partially constructed. In this way, we ensure that bricks can only attach to partially built assemblies containing a seed brick, mimicking the seeded nature of an aTAM system. Additionally, we divide instances of glues into *inward* and *outward* glue sets, such that an outward glue g can only attach to an inward glue of the same type. Throughout the assembly process, the invariant that all exposed glues in any assembly containing a seed brick are outward glues is maintained; this prevents partially built seeded assemblies from attaching to each other. An example of the construction in which 3×3 bricks, 3×1 mortar rectangles, and individual mortar tiles attach to form 5×5 supertiles can be seen in Figure 4.

► **Theorem 4.1.** *Any aTAM system at $\tau \geq 4$ can be simulated by a 2HAM system at $\tau = 4$.*

4.3 Simulating aTAM at $\tau \in \{1, 2\}$ with 2HAM $\tau = 2$

The construction described in the previous section can be modified to also enable simulating aTAM systems at $\tau = \{1, 2\}$ with the 2HAM at $\tau = 2$ with scale factor 5.

► **Theorem 4.2.** *Any aTAM system at $\tau \in \{1, 2\}$ can be simulated by a 2HAM system at $\tau = 2$.*

4.4 Simulating aTAM at $\tau = 3$ with 2HAM $\tau = 3$

The construction used to simulate the $\tau \geq 4$ aTAM model with the $\tau = 4$ 2HAM model can also be modified to simulate the $\tau = 3$ aTAM model with the $\tau = 3$ 2HAM model. The construction given also simulates the aTAM model under the restriction of planarity (tiles can only attach at locations on the exterior of the assembly).

► **Theorem 4.3.** *Any aTAM system at $\tau = 3$ can be simulated by a 2HAM system at $\tau = 3$.*

5 Verification algorithms for aTAM and 2HAM

In this section, we explore the algorithmic complexities of verifying certain properties of a given (2HAM or aTAM) tile assembly system. Sections 3 and 4 suggest that the 2HAM is at least as (if not perhaps strictly more) powerful than the aTAM. In this section, we show that verifying properties of self-assembly systems in the 2HAM is at least as (if not perhaps strictly more) difficult than verifying properties of aTAM systems.

5.1 Unique assembly verification

A fundamental computational problem in self-assembly is that of deciding whether a given self-assembly system uniquely produces a given assembly. We refer to this problem as the Unique Assembly Verification problem (UAV). The aTAM has enjoyed a polynomial time solution [3] to this problem reaching back to 2002. Fast verification within the aTAM has been of tremendous assistance for self-assembly system designers by allowing for simulators that can quickly spot bugs in tile systems. In contrast, the complexity of UAV for 2HAM systems has been a core open problem since the Palaeolithic era. The results of this paper, thus far, seem to suggest “aTAM = O(2HAM)”, i.e., the 2HAM is, in general, at least as powerful as the aTAM. Thus, it should not be difficult for one to believe that, in general, verifying 2HAM systems should be at least as difficult as verifying aTAM systems. In this section, we show that a general fast verification algorithm is unlikely to exist by showing that the UAV is co-NP-complete.

Our UAV co-NP-complete result applies to temperature $\tau = 2$ systems that utilize at most one step into the third dimension⁴. This result resolves the general question of whether efficient unique assembly verification algorithms exist, but leaves open the possibility of a fast algorithm for the important class of 2D 2HAM self-assembly systems. Further, this result is potentially useful for optimistic algorithm designers in search of such 2D efficient systems in that it points out that any such solution will need to make fundamental use of

⁴ We do not formally define the 3D 2HAM because the generalization from the 2D 2HAM is straightforward. Details of the 3D 2HAM that we use can be found in [8]

References

- 1 Zachary Abel, Nadia Benbernou, Mirela Damian, Erik D. Demaine, Martin L. Demaine, Robin Y. Flatland, Scott D. Kominers, and Robert T. Schweller. Shape replication through self-assembly and rnaase enzymes. In *SODA*, pages 1045–1064, 2010.
- 2 Leonard Adleman, Qi Cheng, Ashish Goel, and Ming-Deh Huang. Running time and program size for self-assembled squares. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 740–748, Hersonissos, Greece, 2001.
- 3 Leonard M. Adleman, Qi Cheng, Ashish Goel, Ming-Deh A. Huang, David Kempe, Pablo Moisset de Espanés, and Paul W. K. Rothmund. Combinatorial optimization problems in self-assembly. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 23–32, 2002.
- 4 Leonard M. Adleman, Jarkko Kari, Lila Kari, Dustin Reishus, and Petr Sosik. The undecidability of the infinite ribbon problem: Implications for computing by self-assembly. *SIAM Journal on Computing*, 38(6):2356–2381, 2009.
- 5 Robert D. Barish, Rebecca Schulman, Paul W. Rothmund, and Erik Winfree. An information-bearing seed for nucleating algorithmic self-assembly. *Proceedings of the National Academy of Sciences*, 106(15):6054–6059, March 2009.
- 6 Ho-Lin Chen and David Doty. Parallelism and time in hierarchical self-assembly. In *SODA 2012: Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*.
- 7 Qi Cheng, Gagan Aggarwal, Michael H. Goldwasser, Ming-Yang Kao, Robert T. Schweller, and Pablo Moisset de Espanés. Complexities for generalized models of self-assembly. *SIAM Journal on Computing*, 34:1493–1515, 2005.
- 8 Matthew Cook, Yunhui Fu, and Robert Schweller. Temperature 1 self-assembly: Deterministic assembly in 3d and probabilistic assembly in 2d. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, 2011.
- 9 Erik D. Demaine, Martin L. Demaine, Sándor P. Fekete, Mashhood Ishaque, Eynat Rafalin, Robert T. Schweller, and Diane L. Souvaine. Staged self-assembly: nanomanufacture of arbitrary shapes with $O(1)$ glues. *Natural Computing*, 7(3):347–370, 2008.
- 10 Erik D. Demaine, Matthew J. Patitz, Robert T. Schweller, and Scott M. Summers. Self-assembly of arbitrary shapes using rnaase enzymes: Meeting the kolmogorov bound with small scale factor (extended abstract). In *STACS 2011: Proceedings of the Twenty Eighth International Symposium on Theoretical Aspects of Computer Science*, pages 201–212, 2011.
- 11 David Doty. personal communication, 2011.
- 12 David Doty, Matthew J. Patitz, Dustin Reishus, Robert T. Schweller, and Scott M. Summers. Strong fault-tolerance for self-assembly with fuzzy temperature. In *FOCS 2010: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 417–426. IEEE, 2010.
- 13 Bin Fu, Matthew J. Patitz, Robert T. Schweller, and Robert Sheline. Self-assembly with geometric tiles. In *ICALP (1)*, pages 714–725, 2012.
- 14 James I. Lathrop, Jack H. Lutz, and Scott M. Summers. Strict self-assembly of discrete Sierpinski triangles. *Theoretical Computer Science*, 410:384–405, 2009.
- 15 Paul W. K. Rothmund and Erik Winfree. The program-size complexity of self-assembled squares (extended abstract). In *STOC '00: Proceedings of the thirty-second annual ACM Symposium on Theory of Computing*, pages 459–468, 2000.
- 16 Paul W.K. Rothmund, Nick Papadakis, and Erik Winfree. Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biology*, 2(12):2041–2053, 2004.
- 17 Erik Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology, June 1998.