Natural Computing (2019) 18:527–548
https://doi.org/10.1007/s11047-019-09740-y

Check for updates

Optimal staged self-assembly of linear assemblies

Cameron Chalk[1] · Eric Martinez[2] · Robert Schweller[2] · Luis Vega[2] · Andrew Winslow[3] · Tim Wylie[2]

Published online: 6 May 2019

Abstract

We analyze the complexity of building linear assemblies, sets of linear assemblies, and $\mathcal{O}(1)$-scale general shapes in the staged tile assembly model. For systems with at most b bins and t tile types, we prove that the minimum number of stages to uniquely assemble a $1 \times n$ *line* is $\Theta(\log_t n + \log_b \frac{n}{t} + 1)$. Generalizing to $\mathcal{O}(1) \times n$ lines, we prove the minimum number of stages is $\mathcal{O}(\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t})$ and $\Omega(\frac{\log n - tb - t\log t}{b^2})$. We also obtain similar upper and lower bounds in a model permitting *flexible glues* using non-diagonal glue functions. Next, we consider assembling sets of lines and general shapes using $t = \mathcal{O}(1)$ tile types. We prove that the minimum number of stages needed to assemble a set of k lines of size at most $\mathcal{O}(1) \times n$ is $\mathcal{O}(\frac{k\log n}{b^2} + \frac{k\sqrt{\log n}}{b} + \log\log n)$ and $\Omega(\frac{k\log n}{b^2})$. In the case that $b = \mathcal{O}(\sqrt{k})$, the minimum number of stages is $\Theta(\log n)$. The upper bound in this special case is then used to assemble "hefty" shapes of at least logarithmic edge-length-to-edge-count ratio at $\mathcal{O}(1)$-scale using $\mathcal{O}(\sqrt{k})$ bins and optimal $\mathcal{O}(\log n)$ stages.

Keywords Tile self-assembly · Staged self-assembly · DNA computing · Biocomputing

1 Introduction

Modern technology applications increasingly involve precise design and manufacture of materials and devices at the nanoscale. One approach to nanoscale design is to use *self-assembly*: local interaction rules that direct the aggregation of large numbers of simple units. Seeman (1982) discovered that short strands of DNA whose interactions are controlled by attraction between their base sequences can be programmed to carry out such self-assembly. This approach was subsequently extended both experimentally and theoretically by Winfree (1998), who introduced the *abstract Tile Assembly Model (aTAM)* to describe systems of four-sided planar *tiles* which randomly collide and attach if abutting sides have matching *glues* of sufficient bonding strength. This simple model is computationally universal Winfree (1998) and experimentally capable of complex algorithmic behaviors Evans (2014).

Staged tile assembly Here we study a tile assembly model introduced by Demaine et al. (2008) that permits carrying out assembly in multiple *bins* whose products can be mixed together later, capturing the common experimental technique of decomposing a complex reaction into *stages* of simpler reactions Tikhomirov et al. (2017). This model generalizes the *two-handed* Cannon et al. (2013) or *hierarchical* Chen and Doty (2012) *tile self-assembly*

This research was supported in part by National Science Foundation Grants CCF-1117672, CCF-1555626, and CCF-1817602.

✉ Tim Wylie
timothy.wylie@utrgv.edu

Cameron Chalk
ctchalk@utexas.edu

Eric Martinez
eric.m.martinez02@utrgv.edu

Robert Schweller
robert.schweller@utrgv.edu

Luis Vega
luis.a.vega01@utrgv.edu

Andrew Winslow
andrewwinslow@gmail.com

1 Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA

2 Department of Computer Science, University of Texas Rio Grande Valley, Edinburg, TX, USA

3 San Bruno, CA, USA

Table 1 An overview of old and new results on problems considered in this paper

Bins	Tiles	Upper bound	Lower bound	References
$1 \times n$ lines				
$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\Theta(\log n)$	$\Theta(\log n)$	Cor. 1, Thm. 3 of Demaine et al. (2008)
b	t	$\Theta(\log_t n + \log_b \frac{n}{t} + 1)$	$\Theta(\log_t n + \log_b \frac{n}{t} + 1)$	Theorems 1, 2
$\mathcal{O}(1) \times n$ lines (standard glues)				
1	$n^{\mathcal{O}(1)}$	1	1	Thm. 3.2 of Cheng et al. (2005)
b	t	$\mathcal{O}(\frac{\log n - t\log t - tb}{b^2} + \frac{\log\log b}{\log t})$	$\Omega(\frac{\log n - t\log t - tb}{b^2})$	Theorems 4, 6
$\mathcal{O}(1) \times n$ lines (flexible glues)				
1	$n^{\mathcal{O}(1)}$	1	1	Thm. 3.2 of Cheng et al. (2005)
b	t	$\mathcal{O}(\frac{\log n - t^2 - tb}{b^2} + \frac{\log\log b}{\log t})$	$\Omega(\frac{\log n - t^2 - tb}{b^2})$	Theorems 5, 6
Line sets				
b	$\mathcal{O}(1)$	$\mathcal{O}(\frac{k\sqrt{\log n}}{b} + \frac{k\log n}{b^2} + \log\log n)$	$\Omega(\frac{k\log n}{b^2})$	Theorems 7, 8
$\mathcal{O}(\sqrt{k})$		$\Theta(\log n)$	$\Theta(\log n)$	Theorems 9, 1
Hefty hole-free shapes				
$\mathcal{O}(k)$	$\mathcal{O}(1)$	$\mathcal{O}(\log n)$	$\Omega(\frac{\log n}{k})$	Cor. 1(Demaine et al. (2015)), Thm. 3(Demaine et al. (2008))
$\mathcal{O}(\sqrt{k})$		$\Theta(\log n)$	$\Theta(\log n)$	Theorems 10, 11

Variables t and b denote resource constraints on tile types and bins, respectively. For line sets, k denotes the number of lines in the set, while n denotes the length of the longest line. For general shapes, k denotes the number of edges in the shape, while n denotes the edge length of the minimum-diameter bounding square of the shape. A hefty shape is a shape whose edges are all length at least logarithmic in the number of edges

model (2HAM). Unlike the aTAM, in which single tiles attach to a multi-tile seed *assembly*, the 2HAM permits arbitrary pairs of assemblies to attach provided they do so via glues of sufficient strength. Growth without a seed occurs naturally in experimental DNA tile systems Barish et al. (2009), Schulman and Winfree (2007), motivating the study of two-handed models.

Efficient assembly One of the fundamental goals of self-assembly is the design of *efficient* systems that assemble given shapes or patterns. Staged systems have three combinatorial measures of efficiency: the number of tile types (*tile complexity*), the maximum number of bins used in any stage (*bin complexity*), and the number of stages of the system (*stage complexity*). Numerous constructions of efficient staged systems that assemble given shapes Demaine et al. (2008), Demaine et al. (2015) and patterns Demaine et al. (2013), Winslow (2015) have been given. Here, we give new, more efficient constructions for assembling height-1 and height-$\mathcal{O}(1)$ rectangles called *lines*, sets of such lines, and *hefty* general shapes of sufficient edge-length-to-edge-count ratio. The results are summarized in Table 1 and described below.

Assembling $1 \times n$ lines The construction of lines is often used as a subroutine in the assembly of more complex shapes Demaine et al. (2008, 2015) or as a simple benchmark shape Adleman et al. (2001), Chandran et al. (2012). In the 2HAM, assembling a $1 \times n$ line requires n tile types, and thus as a corollary, staged systems with 1 bin, 1 stage, and n tile types assemble $1 \times n$ lines.

If $\mathcal{O}(1)$ bins and $\mathcal{O}(\log n)$ stages are permitted, then $\mathcal{O}(1)$ tile types suffice Demaine et al. (2008), demonstrating a trade-off between two measures of staged system complexity. However, no general trade-off relating all three complexity measures were known prior to this work for assembling $1 \times n$ lines. Here we obtain tight upper and lower bounds that completely characterize the trade-off: for systems of at most t tile types and b bins, the minimum number of stages needed to assemble any $1 \times n$ line is $\Theta(\log_t n + \log_b \frac{n}{t} + 1)$ (Theorems 1 and 2).

A precursor to the upper bound construction was used to generate a set of gadgets to achieve the primary results in Chalk et al. (2018). The lower bound approach (Theorem 2) is novel and is not information-theoretic. As a result, it holds for all n rather than almost all n, a common limitation of information-theoretic lower bounds in tile self-assembly.

Assembling $\mathcal{O}(1) \times n$ lines In the 2HAM, $\mathcal{O}(1) \times n$ lines can be assembled using $n^{\mathcal{O}(1)}$ tile types Cheng et al. (2005),[1] but a lower bound exceeding $\Omega(\frac{\log n}{\log\log n})$ remains open. The assembly of $\mathcal{O}(1) \times n$ lines has not been studied explicitly in the staged model, however some constructions

[1] The result is given for the aTAM in Cheng et al. (2005) but the same tile set at temperature 2 in the 2HAM behaves identically.

of Demaine et al. (2015) utilize $\mathcal{O}(1) \times n$ line construction as a subroutine.

We give staged systems that use t tile types and b bins that assemble $\mathcal{O}(1) \times n$ lines in $\mathcal{O}(\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t})$ stages (Theorem 4) and prove that for almost all n, $\Omega(\frac{\log n - tb - t\log t}{b^2})$ stages are required (Theorem 6). The upper bound implies a number of new results, including the assembly of $\mathcal{O}(1) \times n$ lines by systems with $\mathcal{O}(1)$ bins, $\mathcal{O}(1)$ stages, and $\mathcal{O}(\frac{\log n}{\log\log n})$ tile types, beating our lower bound of $\Omega(\log n)$ tile types for $1 \times n$ lines (Theorem 2).

This result utilizes the *bit-pad* gadget of Chalk et al. (2018), and the construction of this pad is the bottleneck for the complexity we achieve. Used naively, this bit-pad gadget can be used to assemble $\mathcal{O}(\log n) \times n$ rectangles within the stated complexity. Here, we combine the bit-pad gadget with a novel "sideways" counter to reduce the rectangle height from $\mathcal{O}(\log n)$ to $\mathcal{O}(1)$. This counter involves a non-deterministic guessing strategy for copying sets of $\log n$ bits through $\mathcal{O}(1)$-height regions, "deactivating" incorrect copies. This technique solves a common difficulty in assembling shapes with narrow regions of low "geometric bandwidth" Adleman et al. (2009), Cheng et al. (2005) and may have other applications in two-handed self-assembly.

Although we show how to construct $\mathcal{O}(1) \times n$ linear assemblies, we acknowledge that in any implementation, this construction method becomes impractical. The non-deterministic guessing strategy creates an exponential amount of garbage, which is not a problem in the model due to the definition, but in reality would be infeasible. This same exploitable feature also exists in the 2HAM Schweller et al. (2017), and motivates work towards a modification of the model or finding solutions with polynomial or constant sized garbage.

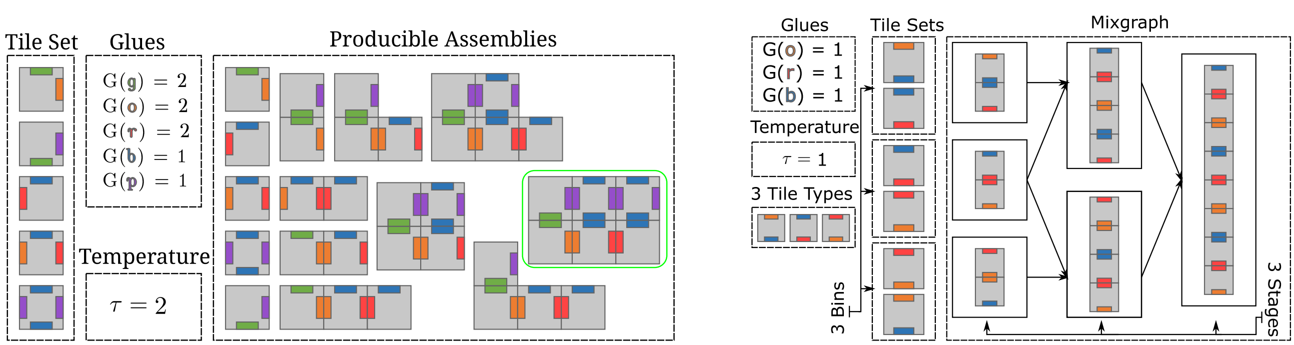
Assembling $\mathcal{O}(1) \times n$ line sets and general shapes Finally, we consider constructing a set of k $\mathcal{O}(1)$-height lines of differing lengths up to n, in service of general shape construction. The first result is a b-bin, $\mathcal{O}(\frac{k\log n}{b^2} + \frac{k\sqrt{\log n}}{b} + \log\log n)$-stage, $\mathcal{O}(1)$-tile system for assembling any such set of lines (Theorem 7). This is complemented by a lower bound of $\Omega(\frac{k\log n}{b^2})$ (Theorem 8), optimal within an additive $\mathcal{O}(\log\log n)$ factor for small b.

In the special case of systems with $\mathcal{O}(\sqrt{k})$ bins and $\mathcal{O}(1)$ tile types, we give a tight bound of $\Theta(\log n)$ stages (Theorem 9 and Corollary 1). We then use the upper bound to efficiently assemble *hefty* shapes whose edge lengths are at least logarithmic in the number of edges with a $\mathcal{O}(1)$ scale factor increase. This small scale factor contrasts with the results of Chalk et al. (2018), where more efficient assembly of shapes is obtained, but with unbounded scale factor.

We also prove that any such shape can be assembled by a system with $\mathcal{O}(1)$ tile types, $\mathcal{O}(\sqrt{k})$ bins, and $\mathcal{O}(\log n)$ stages (Theorem 10), optimal for nearly every choice of k and n (Theorem 11) and giving an affirmative answer to a question of Demaine et al. (2015).

2 The staged self-assembly model

Here, we give a technical introduction to the two-handed tile assembly model (2HAM) and the staged self-assembly model. The *two-handed tile assembly model* is a model of tile-based assembly processes in which large assemblies can combine freely, in contrast to the well-studied aTAM that limits assembly to single-tile addition to a growing seed assembly. An example system is shown in Fig. 1a.



(a) 2HAM Example **(b)** Staged Self-Assembly Example

Fig. 1 **a** A 2HAM example that uniquely builds a 2×3 rectangle. The top 4 tiles in the tile set all combine with strength-2 glues building the 'L' shape. The tile with blue and purple glues needs two tiles to cooperatively bind to the assembly with strength 2. All possible producibles are shown with the terminal assembly highlighted. **b** A simple staged self-assembly example. The system has 3 bins and 3 stages, as shown in the mixgraph. There are three tiles in our system that we assign to bins as desired. From each stage only the terminal assemblies are added to the next stage. The result of this system is the assembly shown in the bin in stage 3. (Color figure online)

The *staged self-assembly model* is a generalization of the 2HAM in which the terminal assemblies of one 2HAM system can be used, in place of single tiles, as the input assemblies of another 2HAM system. Each system exists in a separate *bin*, and the terminal assemblies of a set of bins can be combined as the input assemblies to another bin in the subsequent *stage*. A staged system then consists of a mixing "graph" that defines which bins' contents are mixed into each bin in the subsequent stage. Figure 1b shows a small example system.

Tiles A *tile* is a non-rotatable unit square with each edge labeled with a *glue* from a set Σ. Each pair of glues $g_1, g_2 \in \Sigma$ has a non-negative integer *strength* $\mathrm{str}(g_1, g_2)$. Every set Σ contains a special *null glue* whose strength with every other glue is 0. If $\mathrm{str}(g_1, g_2) \neq 0$ for some $g_1 \neq g_2$, then the glues are *flexible*. Unless otherwise stated, we assume that glues are not flexible.

Configurations, bond graphs, and stability A *configuration* is a partial function $A : \mathbb{Z}^2 \to T$ for some set of tiles T, i.e. an arrangement of tiles on a square grid. For a given configuration A, define the *bond graph* G_A to be the weighted grid graph in which each element of $\mathrm{dom}(A)$ is a vertex, and the weight of the edge between a pair of tiles is equal to the strength of the coincident glue pair. A configuration is said to be τ-*stable* for positive integer τ if every edge cut of G_A has strength at least τ, and is τ-*unstable* otherwise.

Assemblies For a configuration A and vector $\mathbf{u} = \langle u_x, u_y \rangle$ with $u_x, u_y \in \mathbb{Z}^2$, $A + \mathbf{u}$ denotes the configuration $A \circ f$, where $f(x, y) = (x + u_x, y + u_y)$. For two configurations A and B, B is a *translation* of A, written $B \simeq A$, provided that $B = A + \mathbf{u}$ for some vector $\mathbf{u}$. For a configuration A, the *assembly* of A is the set $\tilde{A} = \{B : B \simeq A\}$. An assembly $\tilde{A}$ is a *subassembly* of an assembly $\tilde{B}$, denoted $\tilde{A} \sqsubseteq \tilde{B}$, provided that there exists an $A \in \tilde{A}$ and $B \in \tilde{B}$ such that $A \subseteq B$. An assembly is τ-*stable* provided the configurations it contains are τ-stable. Assemblies $\tilde{A}$ and $\tilde{B}$ are τ-*combinable* into an assembly $\tilde{C}$ provided there exist $A \in \tilde{A}$, $B \in \tilde{B}$, and $C \in \tilde{C}$ such that $A \cup B = C$, $A \cap B = \varnothing$, and $\tilde{C}$ is τ-stable.

Two-handed assembly and bins We define the assembly process in terms of bins. A *bin* is an ordered tuple (S, τ) where S is a set of *initial* assemblies and τ is a positive integer parameter called the *temperature*. For a bin (S, τ), the set of *produced* assemblies $P'_{(S,\tau)}$ is defined recursively as follows:

1. $S \subseteq P'_{(S,\tau)}$.
2. If $A, B \in P'_{(S,\tau)}$ are τ-combinable into C, then $C \in P'_{(S,\tau)}$.

A produced assembly is *terminal* provided it is not τ-combinable with any other producible assembly, and the set of all terminal assemblies of a bin (S, τ) is denoted $P_{(S,\tau)}$. Intuitively, $P'_{(S,\tau)}$ represents the set of all possible supertiles that can self-assemble from the initial set S, whereas $P_{(S,\tau)}$ represents only the set of supertiles that cannot grow any further.

The assemblies in $P_{(S,\tau)}$ are *uniquely produced* iff for each $x \in P'_{(S,\tau)}$ there exists a corresponding $y \in P_{(S,\tau)}$ such that $x \sqsubseteq y$. Thus unique production implies that every producible assembly can be repeatedly combined with others to form an assembly in $P_{(S,\tau)}$.

Staged assembly systems An *r-stage b-bin mix graph M* is an acyclic r-partite digraph consisting of rb vertices $m_{i,j}$ for $1 \le i \le r$ and $1 \le j \le b$, and edges of the form $(m_{i,j}, m_{i+1,j'})$ for some i, j, j'. A *staged assembly system* is a 3-tuple $\langle M_{r,b}, \{T_1, T_2, \ldots, T_b\}, \tau \rangle$ where $M_{r,b}$ is an r-stage b-bin mix graph, T_i is a set of tile types, and τ is an integer temperature parameter.

Given a staged assembly system, for each $1 \le i \le r$, $1 \le j \le b$, we define a corresponding bin $(R_{i,j}, \tau)$ where $R_{i,j}$ is defined as follows:

1. $R_{1,j} = T_j$ (this is a bin in the first stage);
2. For $i \ge 2$, $R_{i,j} = \left(\bigcup_{k:\ (m_{i-1,k}, m_{i,j}) \in M_{r,b}} P_{(R_{(i-1,k)},\tau)} \right)$.

Thus, the jth bin in stage 1 is provided with the initial tile set T_j, and each bin in any subsequent stage receives an initial set of assemblies consisting of the terminally produced assemblies from a subset of the bins in the previous stage as dictated by the edges of the mix graph.[2] The *output* of the staged system is simply the union of all terminal assemblies from each of the bins in the final stage.[3] We say that this set of output assemblies is *uniquely produced* if each bin in the staged system uniquely produces its respective set of terminal assemblies.

Shapes The *shape* of an assembly is the polyomino defined by the tile locations, i.e. $\mathrm{dom}(A)$, and is *scaled by a factor* c by replacing each cell of the polyomino with a $c \times c$ block of cells. A shape is *hole-free* provided it is simply connected.

Since every shape is a polyomino, its boundary consists of unit-length horizontal and vertical line segments. An *edge* of a shape is a maximal contiguous parallel sequence of such segments. A shape with k edges is *hefty* provided each edge has length at least $\frac{4 \log_2 k + 4}{26} = \Omega(\log k)$. A shape S

[2] The original staged model Demaine et al. (2008) only considered $\mathcal{O}(1)$ distinct tile types, and thus for simplicity allowed tiles to be added at any stage. Because systems here may have super-constant tile complexity, we restrict tiles to only be added at the initial stage.

[3] This is a slight modification of the original staged model Demaine et al. (2008) in that the final stage may have multiple bins. However, all of our results apply to both variants of the model.

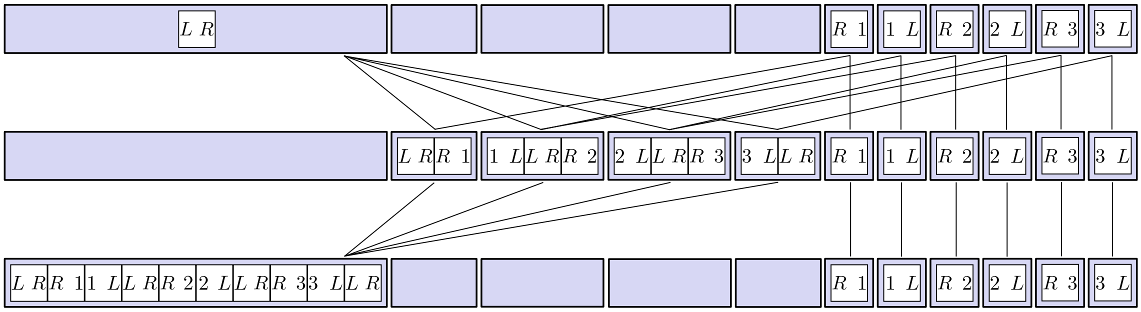


Fig. 2 A high-level example using $t = 7$ tile types and 11 bins. Note that the growing assembly in the third stage's leftmost bin maintains the property that L and R glues are exposed on the left and right identical to the single tile in the first stage's leftmost bin. This two-stage mixing process repeats, each time increasing the length of the assembly in the leftmost bin by a factor of $\Theta(t)$

is an $h \times w$ *line* provided $S = \{y+1, y+2, \ldots, y+h\} \times \{x+1, x+2, \ldots x+w\}$ for some $x, y \in \mathbb{Z}^2$.

3 Assembling $1 \times n$ lines

We start by analyzing the parameterized staged complexity of assembling $1 \times n$ lines using systems with t tile types and b bins. We first provide a high-level explanation and example, then give formal construction details.

In the case of Lemma 1 (when $b \geq \frac{3}{2}t + \frac{5}{2}$), t' copies of a $1 \times \ell$ assembly are assembled into a $1 \times \ell\ t'$ assembly in two stages (initially, $\ell = 1$). An example of this technique for a specific t and b can be seen in Fig. 2. Growing by a factor of t' in $\mathcal{O}(1)$ stages implies $\mathcal{O}(\log_t n)$ stages suffice to assemble $1 \times n$ lines, where n is a power of t'. Since this system generates all powers of t' in intermediate stages, values of n that are not powers of two are handled by keeping a *partial growth bin* where k distinct $1 \times (t')^i$ assemblies are concatenated to a growing assembly each time the ith digit in the base t' expansion of n is k. If Lemma 1 does not apply but $b \geq \frac{t}{2}$, then shrinking t by a factor of 3 and applying Lemma 1 implies $\mathcal{O}(\log_{t/3} n + 1) = \mathcal{O}(\log_t n + 1)$ stages suffice.

Otherwise, $t/2 > b$ and Lemma 2 applies. In this case, the above technique fails because there are too few bins for the t' tiles used to connect t' copies of a $1 \times \ell$ assembly. Instead, the assembly is grown by factors of b' (rather than t') using b' tile types as connectors. The $t' - b'$ tiles not used as connectors create a $1 \times (t' - b')$ assembly that is assigned in the first stage to each of the connector tiles' bins, increasing the length of connectors in the first stage. Growing by a factor of b' in $\mathcal{O}(1)$ stages using assemblies which start at length $t' - b'$ implies $\mathcal{O}(\log_b \frac{n}{t-b} + 1)$ stage complexity. Lengths that are not powers of b' are handled identically as in Lemma 1, but utilizing the base b' (rather than base t') expansion of n. Since $\frac{t}{2} > b$, $\mathcal{O}(\log_b \frac{n}{t-b}) = \mathcal{O}(\log_b \frac{n}{t})$.

The following upper bound follows immediately from combining the construction of Lemmas 1 and 2.

Theorem 1 *There exists a constant c such that for any $b, t, n \in \mathbb{N}$ with $b, t > c$ there exists a staged assembly system with b bins and t tile types whose uniquely produced output is a $1 \times n$ line using $\mathcal{O}(\log_t n + \log_b \frac{n}{t} + 1)$ stages.*[4]

Lemma 1 *For any $b, t, n \in \mathbb{N}$ with $t \geq 5$ and $b \geq \frac{3}{2}t + \frac{5}{2}$, there exists a staged assembly system with b bins and t tile types whose uniquely produced output is a $1 \times n$ line using $\mathcal{O}(\log_t n + 1)$ stages.*

Proof We describe a staged system with b bins and t tiles satisfying the statement. Let $\gamma = \lfloor \frac{t-1}{2} \rfloor$ and let $n' = \lfloor \frac{n}{3} \rfloor$. The initial tile assignment is as follows:

- The *growth bin* (*gb*) contains a tile with L and R glues on its left and right side respectively.
- The *connector bin 1* (*cb*1) contains a tile with R and C glues on its left and right side, respectively.
- The *connector bin 2* (*cb*2) contains a tile with C and L glues on its left and right side, respectively.
- $(\gamma - 1)$ bins are labeled rcb_j (*right connector bin* j) for $j \in \{1, 2, \ldots, \gamma - 1\}$ and each contain a tile type with glues R and j on its left and right edges, respectively.
- Similarly, another $\gamma - 1$ bins are labeled lcb_j and contain tiles with glues j and L on their left and right edges.

An additional $\gamma + 3$ bins begin empty:

- The *incubator bin* (*ib*).
- The *stopper bin* (*sb*).

[4] The "+1" implies the trivial requirement of at least one stage.

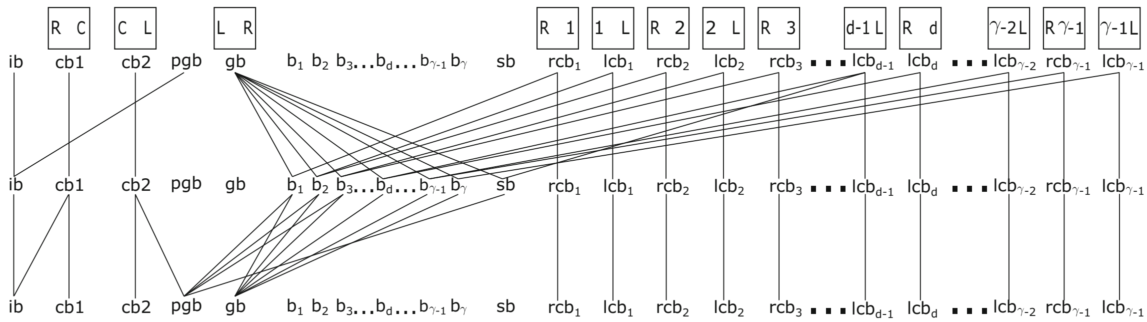


Fig. 3 The two-stage mixing process. Note that b_i for $i \geq d$ do not mix into *pgb*. If a bin does not have an edge to itself from one stage to the next, the bin is emptied

- The *partial growth bin* (*pgb*).
- γ bins labeled $b_1, b_2, \ldots, b_\gamma$.

Repeat the following two-stage mixing process $\log_\gamma n' + 2$ times, for $2\log_\gamma n' + 4$ total stages (see Fig. 3). Let k be the number of times the two-stage process has occurred, and let d_k be the kth digit in the base-γ expansion of n'. Note that the following is just a formal treatment of the technique described in the high-level paragraph at the beginning of the section. In the first stage:

- Mix *gb* into $b_1, b_2, \ldots, b_\gamma$.
- Mix *pgb* into *ib*.
- For each rcb_j, mix rcb_j and lcb_{j-1} into b_j (mix rcb_1 into b_1 as there is no lcb_0, and mix $lcb_{\gamma-1}$ into b_γ).
- Mix *gb* and lcb_{d_k-1} into *sb*.
- Carry *ib*, *cb*1, *cb*2, and all rcb_j and lcb_j to the next stage.

In the second stage:

- Mix *cb*1 into *ib*.
- Mix *cb*2 into *pgb*.
- Mix all b_i into *gb*.
- Mix b_m $(1 \leq m < d_k)$ and *sb* into *pgb*.
- Carry down *ib*, *cb*1, *cb*2, and all rcb_j and lcb_j to the next stage.

Let $L(k)$ be the length of an assembly in the *gb* bin after the kth repetition. Then $L(k) = L(k-1)\gamma + 2(\gamma - 1)$ (γ copies of the assembly in the *gb* bin after the $(k-1)$st repetition plus $2(\gamma - 1)$ connector tiles) with $L(0) = 1$ (the initial tile placed in *gb*). Solving the recurrence, $L(k) = 3\gamma^k - 2$.

Let $PL(k)$ be the length of the assembly in *pgb* after k repetitions of the two-stage process. Attaching d_{k-1} copies of the contents of *gb* (each of length $3\gamma^{k-1} - 2$) into *pgb* using $2(d_{k-1} - 1)$ connector tiles implies:

$$\begin{aligned} PL(k) &= d_{k-1}(3\gamma^{k-1} - 2) + 2(d_{k-1} - 1) \\ &= d_{k-1}3\gamma^{k-1} - 2. \end{aligned}$$

At each repetition of the two-stage process, *pgb* is mixed into *ib*, attaching the assembly in *pgb* to the assembly in *ib* using the 2 tiles from *cb*1 and *cb*2. Let $IL(k)$ be the length of the assembly in *ib* after k repetitions of the two-stage process. Then

$$IL(k) = 3\sum_{i=0}^{k-1} d_i\gamma^i$$

and after $\lceil \log_\gamma n' \rceil + 1$ repetitions, $IL(\lceil \log_\gamma n' \rceil + 1) = 3\sum_{i=0}^{\lceil \log_\gamma n' \rceil} d_i\gamma^i = 3n' = 3\lfloor \frac{n}{3} \rfloor$. Since $\left| n - 3\lfloor \frac{n}{3} \rfloor \right| \leq 2$, at most 2 tiles from *cb*1, *cb*2 can be added to *gb* in two more stages to yield a $1 \times n$ assembly using $2(\gamma - 1) + 3 = 2\gamma + 1 \leq t$ tiles. In total, t bins are used for the t tile types, and $\gamma + 3 = \frac{t-1}{2} + 3$ additional bins for a total of $t + \frac{t-1}{2} + 3 \leq b$ bins. The stage complexity is $2\lceil \log_\gamma n' \rceil + 2 \leq 2\log_{\frac{t-1}{2}} \lfloor \frac{n}{3} \rfloor + 3 = \mathcal{O}(\log_t n)$. $\square$

Lemma 2 *For any $b, t, n \in \mathbb{N}$ with $b > 11$ and $\frac{3}{2}t + \frac{5}{2} > b$, there exists a staged assembly system with b bins and t tile types whose uniquely produced output is a $1 \times n$ line using $\mathcal{O}(\log_b \frac{n}{t-b} + 1)$ stages.*

Proof Let $\beta = \frac{b}{5} - 5$, $r = t - 4\beta$, and $n' = \lfloor \frac{n}{3r} \rfloor$. This construction uses the same approach as Lemma 1 with minor modifications. The extra r tiles are used to initialize every bin with length-r assemblies functionally identical to the single tiles of Lemma 1, e.g., bin *gb* initially contains a $1 \times r$ assembly with an L exposed on its west end and an R on its east end.

This is done by assigning each bin a set of r tiles. Of these tiles, $r - 2$ are generic across all bins and form a $1 \times (r - 2)$ assembly exposing X (left) and Y (right) glues. The remaining two tiles have glues assigned that, when mixed with the $r - 2$ generic tiles, form an assembly with identical left and right glues to the corresponding tile in the bin of the Lemma 1 construction.

Due to bin constraints, only bins $lcb_0, lcb_1 \cdots lcb_{\beta-1}$, $rcb_0, rcb_1 \cdots rcb_{\beta-1}$, $b_0, b_1 \cdots b_\beta$, *ib*, *cb*1, *cb*2, *pgb*, *gb*, *sb* and one new bin, *f*, are used (a total of $3\beta + 8 = 3(\frac{b}{5} - 5) + 8 = \frac{3}{5}b - 7 < b$ bins). Each two-stage mixing process creates and concatenates powers of β rather than γ (from Lemma 1) and will consider the base β expansion of n'.

The two-stage process is repeated $\lceil \log_\beta n' \rceil + 1$ times ($2\lceil \log_\beta n' \rceil + 2$ total stages), where the value of d_k is the kth digit in base-β expansion of n', rather than base-γ as in the proof of Lemma 1. So the length $L(k)$ of an assembly in the gb bin after k repetitions is $L(k) = L(k-1)\beta + 2r(\beta - 1)$ (β copies of the contents of gb plus $2(\beta-1)$ connector assemblies of length r) with $L(0) = r$ (the initial set of r tiles placed in gb). Thus $L(k) = 3r\beta^k - 2r$.

Let $PL(k)$ be the length of the assembly in pgb after k repetitions of the two-stage process. Since d_{k-1} copies of the contents of gb (length $3r\beta^{k-1} - 2r$) are attached in pgb using $2(d_{k-1} - 1)$ connector assemblies of length r from $cb1$ and $cb2$, then:

$$\begin{aligned} PL(k) &= d_{k-1}(3r\beta^{k-1} - 2r) + 2r(d_{k-1} - 1) \\ &= d_{k-1}(3r\beta^{k-1} - 2r) + 2rd_{k-1} - 2r \\ &= d_{k-1}3r\beta^{k-1} - 2r. \end{aligned}$$

Then let $IL(k)$ be the length of the assembly in ib after k repetitions of the two-stage process and so:

$$IL(k) = \sum_{i=0}^{k-1} d_i 3r\beta^i$$

and after $\lceil \log_\beta n' \rceil + 1$ repetitions,

$$IL(\lceil \log_\beta n' \rceil + 1) = 3r \sum_{i=0}^{\lceil \log_\beta n' \rceil} d_i \beta^i = 3rn' = 3r\left\lfloor \frac{n}{3r} \right\rfloor$$

Since $\left| n - 3r\lfloor \frac{n}{3r} \rfloor \right| \le 2r + (r-1)$, the length of the assembly in gb must now be augmented by at most $3r - 1$. Depending on the length discrepancy, mix $cb1$ (length r), $cb1$ and $cb2$ (length $2r$), or neither, along with bin f (the *finisher* bin), initialized with up to $r-1$ from the tiles used to build the $1 \times (r-2)$ midsection of each initial bin assembly plus (possibly) any one additional tile. Combine this assembly with that in gb to reach the desired length n.

In total, $r-2$ tiles are used for the common initial bin $1 \times (r-2)$ assembly, $4(\beta - 1)$ left and right connector tiles, and 6 tiles for gb, $cb1$ and $cb2$ for $4(\beta-1) + 6 + r - 2 = 4\beta + r = t$ total tile types. The stage complexity is

$$\begin{aligned} 2\lceil \log_\beta n' \rceil + 2 &= 2\left\lceil \log_{\frac{b-15}{3}} \left\lfloor \frac{n}{3(t-4\beta)} \right\rfloor \right\rceil + 2 \\ &\le 2\left\lceil \log_{\frac{b-15}{3}} \left\lfloor \frac{n}{3(t-\frac{4}{5}b)} \right\rfloor \right\rceil + 2 \\ &= \mathcal{O}\left(\log_b \frac{n}{t-b}\right). \end{aligned}$$

□

3.1 Lower bound

A lower bound can also be shown for assembling $1 \times n$ lines by proving an equivalent statement: that a system with s stages, b bins, and t tile types can uniquely assemble only lines of length $\mathcal{O}(\min(t^s, tb^s))$.

Theorem 2 *For any $b, t, n \in \mathbb{N}$, a staged system with b bins and t tile types whose uniquely produced output is a $1 \times n$ line must use $\Omega(\log_t n + \log_b \frac{n}{t} + 1)$ stages.*

Proof The bound is equivalent to the statement that a system with s stages, b bins, and t tile types can uniquely assemble only lines of length $\mathcal{O}(\min(t^s, tb^s))$. We prove separately that each of the two quantities is an upper bound.

Upper bound 1: $\mathcal{O}(t^s)$ Consider a bin with an input set of height-1 line assemblies. If any pair of producible assemblies with the same westmost or eastmost tile type can combine in the bin, then an infinite assembly is also assembled and the system cannot uniquely assemble any finite line.

Now suppose creating infinite assemblies is forbidden. Then the length of any terminal assemblies of a stage-1 bin is at most t, and the length of any terminal assembly of a bin is at most t times longer than any input assembly. So the length of any terminal assembly of a stage-s bin is at most t^s.

Upper bound 2: $\mathcal{O}(tb^s)$ For a set of assemblies T, define a subset $C \subset T$ to be *combinable in* T provided that there exists another set of assemblies H such that a bin with input assembly set $T \cup H$ has a unique finite terminal assembly assembled with elements of C. That is, the assemblies of C can be combined into a single finite assembly with the aid of the assemblies of H and without the creation of infinite assemblies.

Observe that a set $C \subset T$ is combinable in T only if for every pair of assemblies with west-east glues pairs (w_1, e_1) and (w_2, e_2), T does not have assemblies with both west-east glue pairs (w_1, e_2) and (w_2, e_1). For the terminal assembly set T of a bin, call $\max\{\sum_{\alpha \in C} |\alpha| : C \text{combinable in} T\}$ the *score* of the bin. Similarly, call the same expression maximized for sets T consisting of a union of terminal assemblies for some subset of the bins of a stage the *score* of the stage. Since a terminal assembly is a singleton combinable set, the score of a stage is an upper bound on the size of any terminal assembly of a bin in that stage. Also, the score of a stage is at most b times the largest score of a bin in the stage, since the condition for a set to be combinable in T is only weakened when T is restricted from the terminal assemblies of multiple bins to those of a single bin.

Consider the score of a bin in stage s with terminal assembly set T. By definition, any combinable subset $C \subset T$ must be assembled out of input assemblies forming a combinable subset of the terminal assemblies from a set of bins of stage $s-1$. Then if each such input assembly appears exactly once in the combinable subset C, the score of the bin is at most the score of the previous stage.

There are two cases for reusing an input assembly β: either two assemblies $\alpha_1, \alpha_2 \in C$ each is assembled with β, or a single assembly is assembled with two copies of β (immediately implying an infinite assembly). In the first case, let α_1 and α_2 have west-east glue pair (w_1, e_1) and (w_2, e_2), respectively. Then T contains two other assemblies with east-west glue pairs (w_1, e_2) (using assemblies in α_1 west of β and in α_2 east of β) and (w_2, e_1) (using the assemblies in α_2 west of β and in α_1 east of β). By a previous claim, such additional pairs are forbidden if α_1, α_2 are both in C. So β cannot be used twice in the assembly of the elements of C, and the score of a bin at stage s is at most the score of stage $s-1$.

In the special case of stage 1, the previous "stage" consists of t bins, each with a separate tile type, and a score of at most t, which is the total length of all terminal assemblies of the stage. Thus, the score of any stage s is at most tb^s. □

4 Assembling $\mathcal{O}(1) \times n$ lines

We now turn our attention to assembling $\mathcal{O}(1) \times n$ lines. Theorem 4 assembles a $\mathcal{O}(1) \times n$ line using a staged system with t tile types, b bins, and $\mathcal{O}(\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t})$ stages, breaking the $\Omega(\log_t n + \log_b \frac{n}{t} + 1)$ lower bound for $1 \times n$ lines.[5] A complementary lower bound of $\Omega(\frac{\log n - tb - t\log t}{b^2})$ for any constant height is given by Theorem 6.

4.1 Special class of $\mathcal{O}(1) \times n$ lines

As a warmup, we describe a simpler construction restricted to an infinite set (but not all) of $\mathcal{O}(1) \times n$ lines. This simpler construction already beats the trivial lower bound of n for $1 \times n$ lines in the aTAM. Details of fine-tuning the termination of the counting, yielding the desired result for all n, are found in the proof of Theorem 4.

Theorem 3 *For any $t, b, n = \Omega(1)$ with $n \in \{i : i = 2^m (2m+3), m \in \mathbb{N}\}$, there exists a temperature-2 staged assembly system with b bins and t tile types whose uniquely produced output is a $\mathcal{O}(1) \times n$ line using $\mathcal{O}(\log\log n)$ stages.*

The construction is broken into four phases, described in the following four subsections. Roughly speaking, the phases are:

1. *Counter gadgets* assemble a horizontal counter that counts from 0 to $2^m - 1$ for some $m \in \mathbb{N}$ with $n = 2^m(2m+3)$. Nondeterminism enables efficiently building all such counter gadgets, but creates many unwanted counter gadgets.
2. *Deactivator gadgets* are assembled. They attach to and *deactivate* unwanted counter gadgets for later disposal.
3. The remaining desired counter gadgets assemble with each other with the help of *gum pads*. The horizontal counter of desired length is assembled.
4. Deactivated counter gadgets are "disposed" by attaching to the bottom of the resulting linear assembly, and the assembly is completed into a rectangle.

4.1.1 Phase 1: assembling counter gadgets

- *Wing gadgets* are rectangular assemblies with geometric bumps on their north surface, where the bumps geometrically encode an *index* in binary using m bits (Fig. 4a).
- A wing gadget *has index* i provided it geometrically encodes a binary string representing i, and all *m-bit* wing gadgets are nondeterministically built using $\mathcal{O}(1)$ tiles, $\mathcal{O}(1)$ bins, and $\mathcal{O}(\log m)$ stages using the mix-graph shown in Fig. 4b.
- Two wing gadgets are nondeterministically brought together with $\mathcal{O}(1)$-size assemblies to form *counter gadgets*, as shown in Fig. 4c.

Counter gadgets are constructed out of *wing gadgets* that geometrically represent bit strings. Wing gadgets in turn consist of *bit gadgets* (seen in Fig. 5). A wing gadget is an *m-bit wing gadget with index* i provided it encodes the length-m bit string with value i. Assemble the bit gadgets into wing gadgets following the approach of Demaine et al. (2008) (seen in Fig. 4b), using $\mathcal{O}(1)$ tiles, $\mathcal{O}(1)$ bins, and $\mathcal{O}(\log m)$ stages and bit gadgets with glues as shown in Fig. 4b. Carry out this process for two distinct (*left* and *right*) wing gadgets using distinct sets of $\mathcal{O}(1)$ glues.

Left and right wings are attached non-deterministically to the southwest and northeast surfaces of a vertical line to yield counter gadgets, as seen in Fig. 6. In addition to a geometric encoding on its north surface, the south surface of the right wing (unlike the left wing) has an encoding of its bit string in glues on its south surface (used later when attaching counters).

[5] Note that the first bound is missing the additive constant to ensure at least one stage. There is still a requirement of at least one stage, but '+1' may be insufficient as the term could be negative.

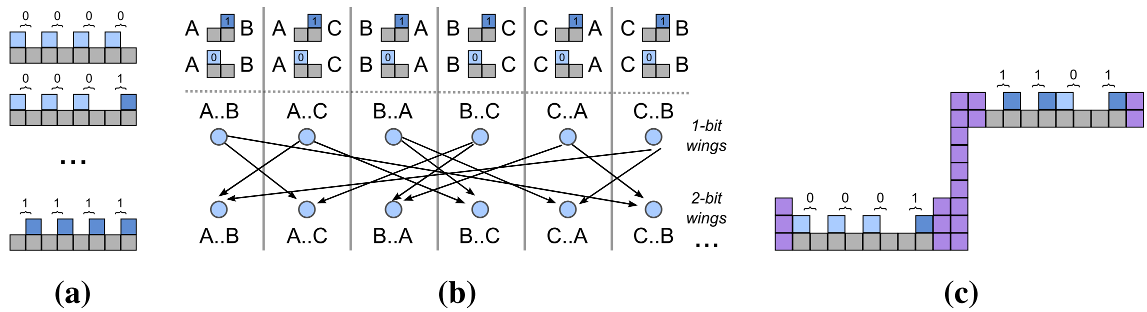


Fig. 4 **a** An example of how 4-bit wing gadgets geometrically encode binary strings. **b** Using $\mathcal{O}(1)$ bins and tile types, the number of bits represented on counter gadgets is doubled every stage. **c** Using vertical lines built from $\mathcal{O}(1)$ tile types, left and right wings are nondeterministically brought together to form a counter gadget

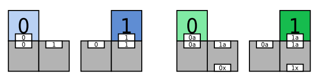


Fig. 5 The two leftmost bit gadgets represent the assemblies used in the construction of the left wing, whereas the other two will be used for the construction of right wings. The right-wing bit gadgets have glues on their southern faces that signify whether the bit represents a 1 or a 0—these will be used to increment the binary representation on the right by 1, in a later stage

4.1.2 Phase 2: deactivating bad counter gadgets

- A *deactivator gadget* detects counter gadgets whose left and right wings do not have the same index and *deactivates* them, preventing their assembly with other counter gadgets in a later stage (Fig. 9a). A deactivator gadget is built by assembling an *error checker* and a *deactivator base*.
- Error checkers (Fig. 7c) are assemblies of $\mathcal{O}(1)$ width and $2m+3$ length that, given an m-bit left wing and right wing gadget, can bind to those gadgets if the binary strings represented by those gadgets differ at any of their m bit locations. These gadgets are built using $\mathcal{O}(1)$ tiles, $\mathcal{O}(1)$ bins, and $\mathcal{O}(\log m)$ stages.
- Alone, error checkers cannot completely guarantee that a counter gadget will not interact with the glues of other assemblies. To deactivate the counter gadgets, error checkers are combined with a *deactivator base* to create our deactivator gadgets (Fig. 8c). The deactivator base is built using $\mathcal{O}(1)$ tiles, $\mathcal{O}(1)$ bins, and $\mathcal{O}(\log m)$ stages.
- Deactivator gadgets are mixed with counter gadgets to deactivate *mismatched* counter gadgets encoding different values on east and west wings (Fig. 9a). Deactivated counter gadgets are "disposed" later.

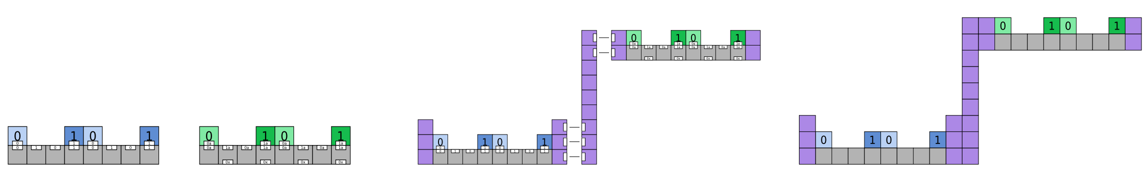


Fig. 6 Non-deterministic assembly of two 4-bit wing gadgets and vertical line to yield a counter gadget. The east and west ends of wing gadgets have extra tiles (purple) to avoid "shifted" interactions between wings. left and right Borders are placed on these lines to avoid shifting errors, and they are attached to the center piece of our counter gadget. (Color figure online)

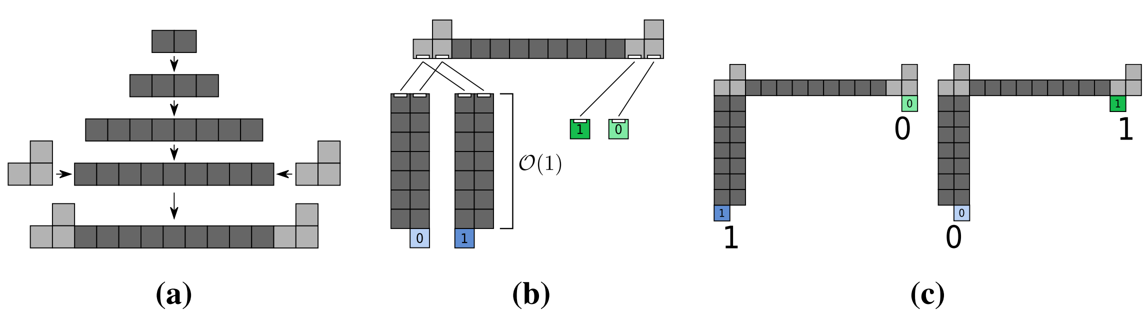


Fig. 7 **a** Assembling an error checker gadget. **b** Left and right *teeth* encoding opposite bit values attach to both ends of the gadget. The tooth heights match the positions of the geometric bits of both wings. **c** The two error checker gadget variants

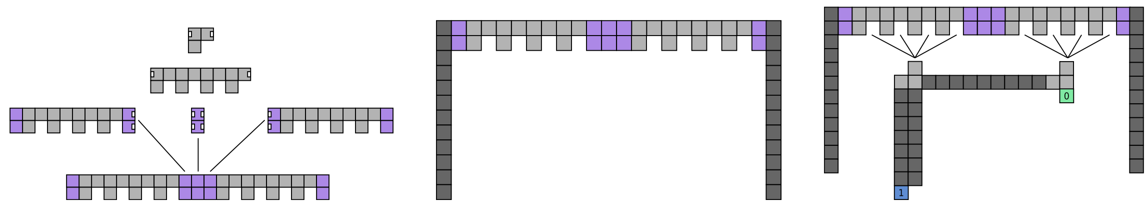


Fig. 8 **a** Assembling the deactivator base. **b** Vertical lines are attached to both sides of the base; the distance between these arms is exactly the length of a counter gadget. **c** The error checker gadget attaching non-deterministically to the base, yielding a deactivator gadget

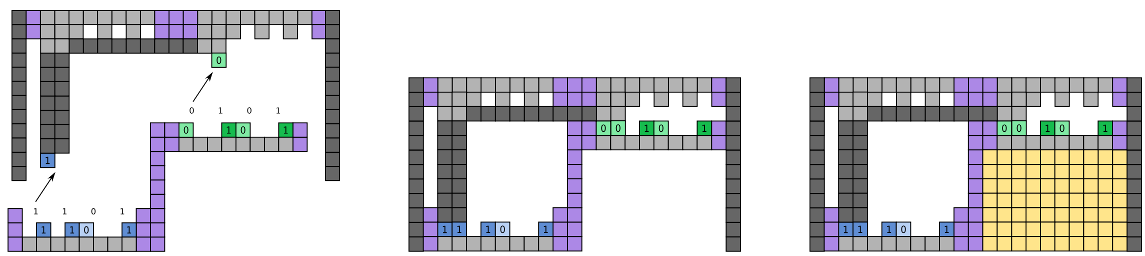


Fig. 9 A completed deactivator gadget attaching to a mismatched counter gadget, after which filler tiles cover the exposed glues that remain on the right wing. The filler tiles begin attaching via a cooperative binding between the counter and deactivator gadgets

Deactivating bins are used to assemble *deactivator* gadgets that attach to "bad" counter gadgets with mismatched left and right wing indices. The north surface placement of the geometric bit string encodings on left and right wings (Fig. 6) allows detection of single-bit (and thus value) mismatches between wings using an *error checker* gadget (seen in Fig. 7). The error checker gadget's length is exactly the distance between pairs of bits with the same power-of-two value in the left and right wings. Both ends of the gadget have geometric designs encoding opposite bit values, thus matching the geometry of a pair of mismatched bit values in the left and right wings. This gadget is easily assembled from lines using $\mathcal{O}(1)$ bins and tiles in $\mathcal{O}(\log m)$ stages, e.g., using the method of Demaine et al. (2008).

Error checker gadgets are non-deterministically combined with a deactivator *base* and vertical lines to yield deactivators. Figure 7 outlines the assembly of deactivators, and Fig. 8 depicts the attachment and "deactivation" of a bad counter gadget. The assembly of deactivator gadgets is easily done using $\mathcal{O}(1)$ bins and tile types, and $\mathcal{O}(\log m)$ stages, where m is the number of bits encoded on the wings of each counter, using previously discussed techniques for assembling lines.

Due to the non-deterministic alignment of deactivator bases and error checker gadgets during assembly, deactivator gadgets for every possible pairs of bits are assembled. Mixing these deactivator gadgets with the set of all counter gadgets causes exactly the bad counter gadgets to have their west, north, and east surfaces "blocked" (see Fig. 9). Adding filler tiles that use cooperative binding between the deactivator and counter gadget "fills in" the region between the deactivator and counter, yielding a rectangular assembly.

4.1.3 Phase 3: line formation

- Counter gadgets that have not been deactivated are mixed with $\mathcal{O}(1)$ *increment tiles* that bind to their right wings, exposing a geometric representation of each wing's binary string, incremented by 1 (Fig. 10a, c).
- *Gum pads* allow a pair of left and right wings on two counter gadgets to attach side-by-side if the indices of the two wings are identical (Fig. 11). Gum pads are built using $\mathcal{O}(1)$ tile types, $\mathcal{O}(1)$ bins, and $\mathcal{O}(\log m)$ stages.
- Gum pads are mixed with the counter gadgets, allowing them to self-assemble into a linear assembly of length n that counts horizontally from 0 to $2^m - 1$.

Next, counter gadgets that have not been deactivated have their right wings incremented and are then assembled into a line. Recall that the glues on the south surfaces of right wings encode the bit string of the wing. Figure 10b shows how $\mathcal{O}(1)$ constant-sized assemblies are used to increment the encoded index by 1 using the standard approach of a carry bit.

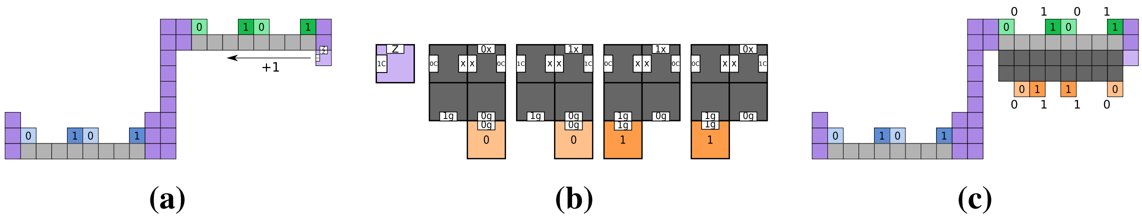


Fig. 10 **a** Increment tiles begin adding *geometric teeth* on the underside of the right wing. **b** The gadgets used to increment. The left-most tile, or increment tile, is what begins the incrementing while the other four are assemblies that will walk along the bottom of the right wing. **c** The geometric teeth on the underside of the right wing. They represent the same number as the top of the right wing after being incremented by one

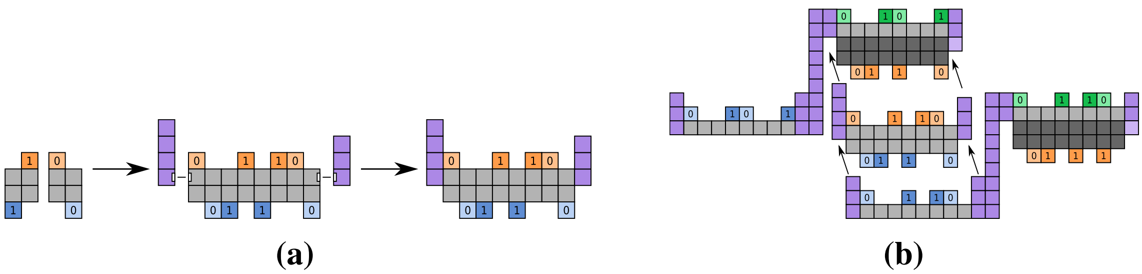


Fig. 11 **a** Gum pad assembly. **b** A gum pad detects matching geometric teeth and adheres two counter gadgets with matching left and right wings together

One detail remains: how to handle incrementing the maximum index. Without modification, the result is to "overflow", incrementing the maximum index to 0 and causing counter assemblies to form an infinitely long assembly. As a fix, we use a single, separate right wing with index 0 to "cap" the right wing with maximum index before incrementing right wing values.

Up to this point, the counter gadgets cannot attach, as left and right wings have no matching glues. Enable attachment by adding *gum pads*: assemblies with identical m-bit binary representations geometrically encoded on their north and south surfaces, but with matching glues for left and right wing gum pads. Figure 11a shows gum pads; they are assembled using the same method as wings.

Due to mirroring, the geometry and glues on the north and south surfaces of the gum pads match right and left wings, respectively. Added east and west borders on right wings, and preexisting borders on left wings (seen also on left wing gum pads in Fig. 11a) prevent shifted attachments. Figure 11b illustrates the interactions of gum pads and counter gadgets.

The deactivated counter gadgets are blocked on their right wings from binding to gum pads, yielding a horizontal counter that stops at the desired length, as shown in Fig. 12.

4.1.4 Phase 4: garbage disposal and finishing

– Deactivated counter gadgets are disposed by attaching to the bottom of the linear assembly, increasing the assembly's width by $\mathcal{O}(1)$, as shown in Fig. 13a.
– A final bin has $\mathcal{O}(1)$ tile types that finish the line by filling any gaps or jagged edges, so that the end result is a rectangle.

To obtain a single, final assembly, attach the *trash* (deactivated bad counter gadgets) to the bottom of the linear assembly. Do so by first attaching unique assemblies to both ends of the line assembly that present a small geometric bump below the line's bottom, geometrically preventing deactivated counters to attach in a way that increases assembly's length beyond n. Figure 13a shows how deactivated counters begin attaching using the cooperative binding created by the leftmost border. From there, other deactivated counter gadgets attach cooperatively to the primary line and prior counter gadgets. Add $\mathcal{O}(1)$ filler tiles to complete in any remaining gaps in the assembly, yielding a $\mathcal{O}(1) \times n$ line.

Stopped counter gadgets are disposed identically to bad counter gadgets. In Phase 2, filler tiles are used to turn deactivated high-value counter gadgets into rectangular assemblies and then in Phase 4, these rectangular assemblies are "disposed" of by attaching them to the bottom of the linear assembly.

We let z be the largest counter value that does not yield an assembly longer than n. More specifically, m is the number of bits in the counter with $2^{m-1}(2(m-1)+3) \leq n < 2^m(2m+3)$, and z is such that $z(2m+3) \leq n < (z+1)(2m+3)$. Thus $z(2m+3)$, the current length of the linear assembly, is shorter than n by $\mathcal{O}(m) = \mathcal{O}(\log n)$. The remaining length is achieved by attaching a $1 \times \mathcal{O}(\log n)$

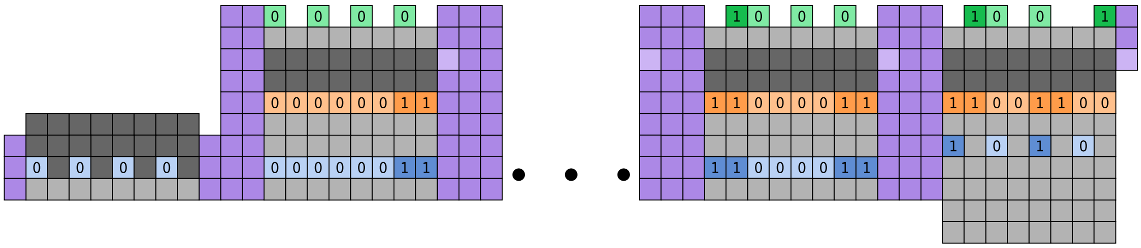
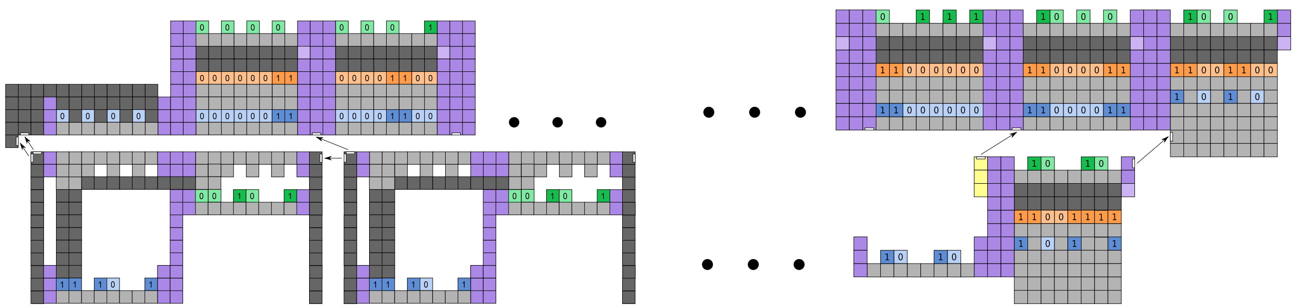


Fig. 12 Counting to a desired value. The last counter gadget is distinguished by the geometric representation of z and binds to a stopper gadget, preventing any further counter gadgets from attaching to its east



(a) **(b)**

Fig. 13 Disposing of trash assemblies. **a** The attachment of "deactivated" counters to the bottom of the linear assembly. $\mathcal{O}(1)$ tiles are added to the westmost edge of the counter. Using these tiles, deactivators can attach to the bottom of the counter. The empty space is filled with $\mathcal{O}(1)$ filler tile types. A geometric bump on the western- and easternmost counter gadgets guides these attachments. **b** Deactivated high-value counter gadgets attaching at the end of the counter, using cooperative binding

line assembled via Theorem 1 using $\mathcal{O}(\log_t \log n + \log_b \frac{\log n}{t} + 1)$ stages.

The remainder of this section describes the modifications of each phase of the previous construction (Theorem 3). One straightforward but important detail is handling the "gap-$\Theta(\log b)$" property of the bit pads assembled by Lemma 3 which implies a spacing of $\Theta(\log b)$ between adjacent bits on the initial bit pad used for the stopper gadget. This spacing is propagated through the construction by adding spacing to all bit gadgets (see Fig. 14). Thus, the linear assembly has spacing as in Fig. 15 before the phases that finish it as a solid $\mathcal{O}(1) \times n$ line.

4.1.5 Complexity

Counter gadgets, deactivator gadgets, and gum pads are all assembled using a common technique borrowed from Demaine et al. (2008) that uses $\mathcal{O}(1)$ tile types and $\mathcal{O}(\log m)$ stages to assemble $\Theta(m)$ assemblies (in $\mathcal{O}(1)$ bins). The same technique is also used to assemble the $\Theta(m)$ lines used in the deactivator gadgets and toothed gum and counter gadget "pads", starting with $\mathcal{O}(1)$ bit gadgets and also using $\mathcal{O}(1)$ bins and $\mathcal{O}(\log m)$ stages. Thus, all aforementioned gadgets can be assembled in parallel using $\mathcal{O}(1)$ tile types, $\mathcal{O}(1)$ bins, and $\mathcal{O}(\log m)$ stages. Since $n = 2^m(2m+3)$, $m = \Theta(\log n)$, and $\mathcal{O}(\log m) = \mathcal{O}(\log \log n)$.

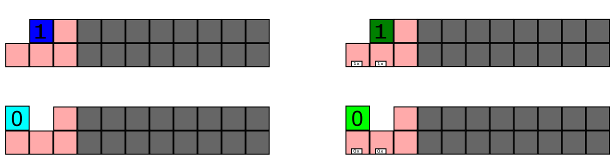


Fig. 14 Bit gadgets with spacing. These assemblies replace the bit gadgets of the previous construction

4.2 Generalizing to all n

The construction of Theorem 3 builds counter gadgets using a horizontal counting method to count from 0 to $2^m - 1$ for any $m \in \mathbb{N}$, yielding assemblies of length $n = 2^m(2m+3)$ for all $m \in \mathbb{N}$. General values of n are achieved by fine-tuning length at two scales: "large scale" via terminating the counter early at a specific value before the desired n and "small scale" via attaching a smaller assembly to reach exactly n from where the counter terminated.

Terminating the counter early is achieved by deactivating "high-value" counter gadgets with values larger than a specified value using *stopper gadgets*, as shown in Fig. 16. Encoding the counter termination value dominates the stage complexity, giving the following result:

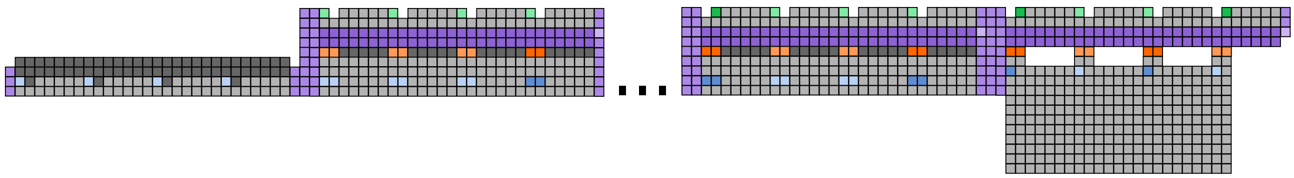


Fig. 15 The linear assembly before finishing. The left wing of the (westernmost) 0-value counter gadget has been capped by a bit string pad of all 0 bits. The right wing of the (easternmost) z-value has a deactivator attached

Theorem 4 *For any* $t, b, n \in \mathbb{N}$ *with* $t, b = \Omega(1)$, *there exists a temperature-2 staged system with b bins and t tile types that assembles a* $\mathcal{O}(1) \times n$ *line using* $\mathcal{O}(\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t})$ *stages.*

Proof Constructing stopper gadgets relies on efficient assembly of a special type of assembly: *width-w*, *gap-f*, *r-bit string pad* is a $w \times f(r-1)+1$ rectangular assembly with r glues of two types, 0 or 1, exposed on the north surface of the rectangle at regular length-f intervals, starting from the western end. The following was proved previously by the authors: □

Lemma 3 (Lemma 3 of Chalk et al. (2018)) *There exists a constant c such that for any* $t, b \in \mathbb{N}$ *with* $t, b > c$ *and any bit string S* of length m, there exists a temperature-2 staged system with b bins and t *tile types whose uniquely produced output is a width-9, gap-* $\log b$, *m-bit string pad encoding S* using $\mathcal{O}(\frac{m - tb - t\log t}{b^2} + \frac{\log\log b}{\log t})$ *stages.*

This efficient method of encoding a bit string is used to assemble such a bit pad encoding z, the largest value the counter can count to without exceeding the desired length n. This bit pad is used to non-deterministically assemble a set of stopper gadgets that deactivate all *high-value* counter gadgets with value at least z. Deactivated counter gadgets are disposed using the same method as previously described in Phase 4.

4.2.1 Phase 2: assembling stopper gadgets

First, a bit string pad encoding z is constructed using Lemma 3. An additional set of $\mathcal{O}(1)$ tile types and a single bin and stage are used to non-deterministically increment the value of this bit string pad by all possible values, yielding bit string pads that encode all values from z to $2^m - 1$ plus "overflowed" values.

Such non-deterministic incrementing is done using a modified version of the standard "zig-zag" approach used in binary counting introduced by the aTAM counter of Rothemund and Winfree (2000). In this counter, single-tile attachments to the north surface compute the result one bit at a time, from least to most significant bit. Each tile attachment converts one of 4 possible input bit and carry bit combinations into an output bit and carry bit pair. Here, 8 tile types are used to augment each combination with a non-deterministic choice of the increment bit value (seen in Fig. 17a).

Because of the $\Theta(\log b)$-length spacing between bits on the bit string pad, filler tile types are used to transmit carry bit values between consecutive bits. If the increment process ends with a carry bit value of 0, then the incremented result did not *overflow* (yield a value larger than can be represented in the available m bits) and the incremented value is in $\{z, z+1, \ldots, 2^m - 1\}$. In this case, $\mathcal{O}(1)$ tile types attach along the north surface in a second row to present the increment m-bit value (Fig. 18a).

If the incremented result did overflow, then the 1-valued carry bit glue initiates grow of a second row that compute the original, unincremented z value (Fig. 18b). This is possible because the north glues presented in the first row encode both the incremented and unincremented (z) values.

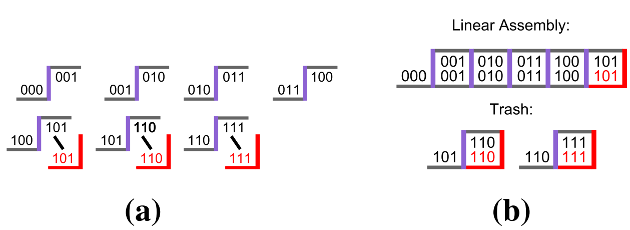


Fig. 16 **a** Stopper gadgets for every number at least 5 assembled and mixed with the counter gadgets. **b** Mixed with gum pads, the counter gadgets assemble, a horizontal counter counting from 0 to 5; with stopped counter gadgets as trash

4.2.2 Phase 2: deactivating high-value counter gadgets

Similar to the deactivation of bad counter gadgets using error checker gadgets, deactivators for high-value counter gadgets are assembled from stopper gadgets and mixed with counter gadgets. Stopper gadgets attach to the right wings of counter gadgets with matching (incremented-by-one, geometrically encoded) values, as shown in Fig. 19a. The resulting deactivated high-value counter gadgets cannot attach to other counter gadgets.

To avoid misaligned attachment of stopper gadgets and counter gadgets, the west end of the stopper gadget has a small part jutting northward. When attached, stopper gadgets jut below the counter gadgets (see Fig. 19b).

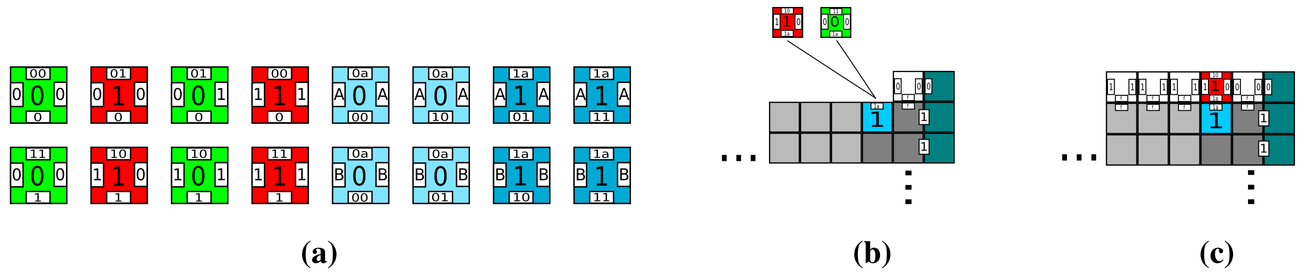


Fig. 17 **a** The tile types used to non-deterministically increment the value of an m-bit string pad encoding z to a value $\geq z$. The 8 leftmost tiles increment z, while the remaining tiles comprise the new encoded value's bits. **b** Each bit is non-deterministically incremented by 0 or 1. **c** Carry bits are handled as in standard "zig-zag" counter constructions

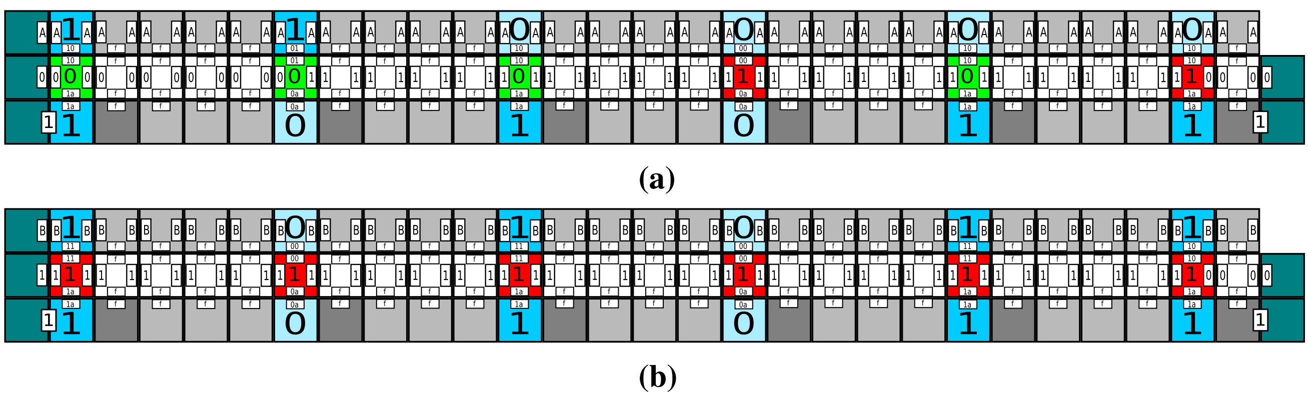


Fig. 18 Two examples of non-deterministically incrementing a bit string pad value z. **a** An increment that does not overflow, yielding a binary value in the range $[z, 2^m - 1]$. **b** An overflowing increment that reverts the binary value back to z

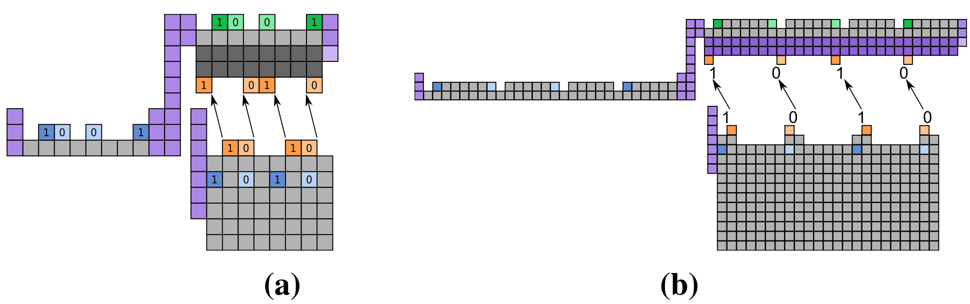


Fig. 19 **a** A stopper gadget attaching to the right wing of a counter gadget, preventing the next counter gadget from attaching. **b** A stopper gadget matching the right wing of a counter. The small part attached to the west end stopper gadget (light purple) prevents misaligned attachment to a counter gadget. (Color figure online)

4.2.3 Phase 5: finishing the line

After deactivating the high-value counter gadgets and proceeding with the assembly of the counter, the assembly constructed has length $\ell = \Theta(zm \log b)$ (recall that each counter gadget has length $\Theta(m \log b)$ due to the spacing added for conformity with the bit string pads used in the stopping gadget). By specifying z (the value of the stopper gadget) as the maximum z such that $\ell \leq n$, the assembly so far is $\mathcal{O}(m \log b) = \mathcal{O}(\log b \log n)$ short of n; call this remaining length n'. Assemble the remaining n' length in two subphases.

For the first subphase, let $b' = \log b$. Employ the same phases 1 through 4 as described above towards assembling a $\mathcal{O}(1) \times n'$ assembly using b' bins and t tile types. The length r of the resulting assembly is $\mathcal{O}(\log n' \log b')$ short of n' (i.e., $n' - r = \mathcal{O}(\log n' \log b')$). The stage complexity of this subphase is $\mathcal{O}(\frac{\log n' - tb' - t \log t}{b'^2} + \frac{\log \log b'}{\log t})$.

In the second subphase, construct a $1 \times (n' - r)$ (i.e. $1 \times \mathcal{O}(\log n' \log b')$) assembly via Theorem 1. The stage complexity of this subphase is $\mathcal{O}(\log_t(\log b' \log n') + \log_b \frac{(\log n' \log b')}{t} + 1)$.

Combine the assemblies constructed via these two subphases to assemble a $\mathcal{O}(1) \times n'$ assembly. Combine this assembly with the $\mathcal{O}(1) \times (n - n')$ assembly constructed via phases 1 through 4 to complete the $\mathcal{O}(1) \times n$ assembly.

4.2.4 Complexity

By Lemma 3, the cost to assemble the initial bit string pad used for the stopper gadgets requires $\mathcal{O}(\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t})$ stages. The remainder of this section argues that this stage complexity is the stage complexity of the entire system; i.e., no other phase requires more stages than this.

First, note the stage complexity of the first subphase of phase 5 is asymptotically less than the stage complexity of assembling the stopper gadgets in phase 2; specifically, $\mathcal{O}\left(\frac{\log\log n \log b - t\log b - t\log t}{\log^2 b} + \frac{\log\log\log b}{\log t}\right) = \mathcal{O}\left(\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t}\right)$.

Next, we upper bound b, the number of bins of the system. Intuitively, above a certain value, more bins do not help reduce the stage complexity. Specifically, if $b = \Omega(\log^{\frac{1}{2}} n)$ the stage complexity of constructing the stopper gadget $\mathcal{O}(\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t})$ increases with b. Therefore, it suffices to consider only the case where $b = \mathcal{O}(\log^{\frac{1}{2}} n)$.

With $b = \mathcal{O}(\log^{\frac{1}{2}} n)$, note that $n' = \log b \log n = \mathcal{O}(\log\log n \log n)$, $b' = \log b = \mathcal{O}(\log\log n)$, and $\log n' \log b' = \mathcal{O}(\log\log\log n \log\log n)$. Then, the stage complexity of the second subphase of phase 5 is

$$\begin{aligned}
&\mathcal{O}\left(\log_t(\log n' \log b') + \log_b \frac{(\log n' \log b')}{t} + 1\right)\\
&= \mathcal{O}\Big(\log_t(\log\log\log n \log\log n).\\
&\quad + \log_b \frac{(\log\log\log n \log\log n)}{t} + 1\Big)\\
&= \mathcal{O}\Big(\log_t(\log\log\log n \log\log n).\\
&\quad + \log_b \frac{(\log\log\log n \log\log n)}{t}\Big)\\
&= \mathcal{O}\Big(\log_t(\log\log\log n) + \log_t(\log\log n).\\
&\quad + \log_b \frac{(\log\log\log n \log\log n)}{t}\Big)\\
&= \mathcal{O}\left(\frac{\log\log\log n}{\log t} + \log_b(\log\log\log n \log\log n) - \log_b t\right)\\
&= \mathcal{O}\Big(\frac{\log\log\log n}{\log t} + \log_b(\log\log\log n)\\
&\quad + \log_b(\log\log n) - \log_b t\Big)\\
&= \mathcal{O}\left(\frac{\log\log\log n}{\log t} + \frac{\log\log\log n}{\log b} - \frac{\log t}{\log b}\right).
\end{aligned}$$

Additionally, spacing the bit gadgets (Fig. 14) to conform with the $\Theta(\log b)$ gap of the stopper gadgets requires the assembly of $1 \times \mathcal{O}(\log b)$ lines via Theorem 1 using $\mathcal{O}(\log_t \log b + \log_b \frac{\log b}{t} + 1)$ stages. Using $b = \mathcal{O}(\log^{\frac{1}{2}} n)$ bins, this stage complexity is

$$\begin{aligned}
&\mathcal{O}\left(\log_t \log\log n + \log_b \frac{\log\log n}{t} + 1\right)\\
&\quad = \mathcal{O}\left(\frac{\log\log\log n}{\log t} + \frac{\log\log\log n}{\log b} - \frac{\log t}{\log b}\right)
\end{aligned}$$

Finally, we analyze cases of values of t and b to prove that

$$\begin{aligned}
&\mathcal{O}\Big(\frac{\log\log\log n}{\log t} + \frac{\log\log\log n}{\log b} - \frac{\log t}{\log b}\\
&\quad + \frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t}\Big)
\end{aligned}$$

is always $\mathcal{O}(\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t})$. That is, the stage complexity of assembling the stopper gadgets dominates the stage complexity of the second subphase of phase 5 and assembling the spacing for the bit gadgets.

- $\mathbf{t \geq \log\log n}$: in this case, $\mathcal{O}\left(\frac{\log\log\log n}{\log t} + \frac{\log\log\log n}{\log b} - \frac{\log t}{\log b}\right) = \mathcal{O}(1)$, while the rightmost two terms are $\Omega(1)$.
- $\mathbf{t, b \leq \log^{\frac{1}{3}} n}$: in this case, $\frac{\log\log\log n}{\log t} + \frac{\log\log\log n}{\log b} - \frac{\log t}{\log b} = \mathcal{O}(\log\log\log n)$, while the rightmost two terms are $\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t} = \Omega(\frac{\log n - tb - t\log t}{b^2}) = \Omega(\log^{\frac{1}{3}} n)$.
- $\mathbf{t \leq \log\log n}$, $\log^{\frac{1}{3}} \mathbf{n} \leq \mathbf{b} \leq \log^{\frac{1}{2}} \mathbf{n}$: in this case, $\frac{\log\log\log n}{\log t} + \frac{\log\log\log n}{\log b} - \frac{\log t}{\log b} = \mathcal{O}\left(\frac{\log\log\log n}{\log t}\right)$, while the rightmost two terms are $\frac{\log n - tb - t\log t}{b^2} + \frac{\log\log b}{\log t} = \Omega\left(\frac{\log\log b}{\log t}\right) = \Omega\left(\frac{\log\log\log n}{\log t}\right)$.

$\square$

Lemma 19 of Chalk et al. (2018) is a strengthened version of Lemma 3 for the flexible glue model, where non-matching glues can have positive attraction strength. Reducing the number of stages used for bit string pad via this construction gives a similarly improved result:

Theorem 5 *There exists a constant c such that for any $t, b, n \in \mathbb{N}$ with $t, b > c$, there exists a temperature-2 staged system with b bins and t tile types whose uniquely produced output is a $\mathcal{O}(1) \times n$ line using $\mathcal{O}(\frac{\log n - tb - t^2}{b^2} + \frac{\log\log b}{\log t})$ stages.*

4.3 Lower bounds for $\mathcal{O}(1) \times n$ lines

Lower bounds for assembling $\mathcal{O}(1) \times n$ lines are obtained using information-theoretic arguments based on combining the bound on information content from Chalk et al. (2018)

with the lower bound of $\lceil \log_2 n \rceil$ on the number of bits needed to specify n for almost all n:

Lemma 4 (Lemma 3.1 of Chalk et al. (2018)) *A staged system of fixed temperature τ with b bins, s stages, and t tile types can be specified using $\mathcal{O}(t \log t + sb^2 + tb)$ bits. Such a system with flexible glues can be specified using $\mathcal{O}(t^2 + sb^2 + tb)$ bits.*

Theorem 6 For any $b, t \in \mathbb{N}$ *and almost all* $n \in \mathbb{N}$, *any staged self-assembly system with b bins and t tile types and uniquely assembles a* $\mathcal{O}(1) \times n$ *line must use* $\Omega(\frac{\log n - tb - t \log t}{b^2})$ *stages, or* $\Omega(\frac{\log n - tb - t^2}{b^2})$ *stages if flexible glues are permitted.*

Proof For almost all $n \in \mathbb{N}$, n requires $\lceil \log_2 n \rceil$ bits to specify. Then by Lemma 4, uniquely assembling a $\mathcal{O}(1) \times n$ line for almost all n is only possible using a number of stages s such that $\log n = \mathcal{O}(sb^2 + t \log t + tb)$. Thus $s = \Omega(\frac{\log n - tb - t \log t}{b^2})$. For systems with flexible glues, Lemma 4 implies that $\log n = \mathcal{O}(sb^2 + t^2 + tb)$ and thus $s = \Omega(\frac{\log n - tb - t^2}{b^2})$. □

5 Assembling $\mathcal{O}(1) \times n$ line sets

Now we consider extending the construction of a $\mathcal{O}(1) \times n$ line to a set of k such lines, working towards the construction of hefty shapes in Sect. 6. The first upper bound construction uses parallel instances of the Theorem 4 construction to assemble multiple lines in parallel with a comparable number of stages.

Theorem 7 *Let* $L = \{n_1, \ldots, n_k\} \subseteq \mathbb{N}$ *with* $n = \max(L)$. *There exists a staged assembly system with* $\mathcal{O}(1)$ *tile types, b bins, and* $\mathcal{O}(\frac{k\sqrt{\log n}}{b} + \frac{k \log n}{b^2} + \log\log n)$ *stages whose uniquely produced output is a set of* $\mathcal{O}(1) \times n_i$ *lines for all* $n_i \in L$.

Proof The approach is to apply the $\mathcal{O}(1) \times n$ construction (with $\mathcal{O}(1)$ tile types) repeatedly to each of the k lines. Note that the $\frac{\log\log b}{\log t}$ additive term of the stage complexity of Theorem 4 is due to the creation of $\mathcal{O}(b)$ assemblies - for Theorem 7 these $\mathcal{O}(b)$ assemblies can be used across the k line segments. So the cost is incurred only once, and is determined by the largest number of bins allotted to any single line - at most $\mathcal{O}(\log\log n)$.

Let $r \in \mathbb{N}$ and b' be a constant fraction of b such that r divides b'. Reserve the remaining $b - b' = \Omega(b)$ bins for additional needed machinery. Divide the b' bins up into r sets of $\frac{b'}{r}$ bins each and use these sets to assemble r lines of length $n_1, n_2, \ldots, n_r$ in $\mathcal{O}(r^2 \frac{\log n}{b^2} + 1)$ stages by applying Theorem 4. Repeat $\lceil \frac{k}{r} \rceil$ times until all k lines are constructed. The total stages used in this process is $\mathcal{O}((r^2 \frac{\log n}{b^2} + 1)(\frac{k}{r} + 1))$.

Selecting $r = \lfloor \frac{b}{\sqrt{\log n}} + 1 \rfloor$ implies $\mathcal{O}(\frac{k\sqrt{\log n}}{b} + \frac{k \log n}{b^2} + 1)$ total stages are used. Adding the one-time cost of $\mathcal{O}(\log \frac{b}{r}) = \mathcal{O}(\log\log n)$ stages discussed above gives $\mathcal{O}(\frac{k\sqrt{\log n}}{b} + \frac{k \log n}{b^2} + \log\log n)$ stages. □

Theorem 8 *Let* $L = \{n_1, \ldots, n_k\} \subseteq \mathbb{N}$ *with* $n = \max(L)$. *For almost all L, any staged self-assembly system with* $\mathcal{O}(1)$ *tile types and b bins that assembles* $\mathcal{O}(1) \times n_i$ *lines for all* $n_i \in L$ *has* $\Omega(\frac{k \log n}{b^2})$ *stages.*

Proof There are at least $(n/2)^k$ choices of L limited to those with $n/2 \leq n_i \leq n$ for all i. So almost all choices of L require $\Omega(\log{(n/2)^k}) = \Omega(k \log n)$ bits. By Lemma 4, any staged system with $\mathcal{O}(1)$ tile types and b bins can be specified with $\mathcal{O}(sb^2)$ bits. So for almost all choices of L, $s = \Omega(\frac{k \log n}{b^2})$ stages are needed. □

In the case that $b = \mathcal{O}(\sqrt{\log n})$, the prior two theorems are tight up to additive terms. However, as b increases, the "crazy mixing" approach Demaine et al. (2008) used in the modular construction of Theorem 7 fails to utilize the growing number of possible mix graphs. The next construction achieves optimal stage complexity for large bin counts, specifically bin counts scaling with k:

Theorem 9 *Let* $L = \{n_1, \ldots, n_k\} \subseteq \mathbb{N}$ *with* $n = \max(L)$. *There exists a staged self-assembly system with* $\mathcal{O}(1)$ *tile types,* $\mathcal{O}(\sqrt{k})$ *bins, and* $\mathcal{O}(\log n)$ *stages that assembles* $\mathcal{O}(1) \times n_i$ *lines for all* $n_i \in L$.

Proof Every positive integer n_i can be represented as a unique sum of distinct powers of two corresponding to the 1-value bits in the binary representation of n_i. Here, a *target line* of length n_i is assembled by concatenating *power-two* assemblies with distinct power-of-two lengths. Each n_i is assigned a *label* in $\{0, 1, \ldots, k-1\}$. This assembly technique occurs in rounds, and in each round i the following occurs:

1. a power-two assembly of length 2^i is assembled by combining two power-two assemblies of length 2^{i-1}, which were assembled in the previous round
2. in n_i's binary representation, if the ith bit has value 1, an assembly encoding n_i's label is assembled and attached to the power-two assembly of length 2^i
3. the now-labeled power-two assemblies are mixed into a bin with the labeled power-two assemblies from the previous rounds

The power-two assemblies only attach to power-two assemblies from previous rounds if their labels match, resulting in k labeled lines, each assembled by concatenating power-two assemblies only when the power of two is included in the unique sum of distinct powers of two for the n_i which had that label assigned.

Primitive gadgets In order to label target assemblies and power-two assemblies, use assemblies of length $2\lceil \log k \rceil$ that encode numbers in $\{0, 1, \ldots, k-1\}$ using geometry reflecting the binary representation of the number in $\log_2 k$ bits. If each of the k labels could be stored in their own bin, then choosing a subset of labels to attach to the power-two assemblies is easily achieved via mixing a subset of bins into the bin containing the power-two assemblies. However, to limit the system to $\mathcal{O}(\sqrt{k})$ bins, $\mathcal{O}(\sqrt{k})$ bins hold labels that encode the $(\log_2 k)/2$-bit numbers $0, 1, \ldots, \sqrt{k}-1$ and have a common glue on their east ends; a second, nearly identical set of bins and labels differ only in presenting the same common glue on their west ends instead. Then, in each round, labels encoding any subset $S \subseteq \{0, 1, \ldots, k-1\}$ assemble in $\mathcal{O}(\sqrt{k})$ bins in the following way: for each length-$\frac{\log k}{2}$ prefix of an integer in S (i.e., a number in $\{0, 1, \ldots, k-1\}$) and set of length-$\frac{\log k}{2}$ suffixes of integers in S with this prefix, mix the corresponding prefix and suffix assemblies into a common bin. Before the first round, assemble all $\sqrt{k}$ label halves of length $\log(\sqrt{k})$ in $\log(\sqrt{k})$ stages, with each label half in one of $\sqrt{k}$ bins (see Fig. 20a).

Also construct two more gadget types involving the geometric labels: *activator* assemblies and *converter* assemblies. Both types read and extend exposed labels and are built in the same fashion as labels: combining all "halves" (shown in Fig. 20b). Activators simply copy labels, exposing the same label on both surfaces, while converter assemblies encode different binary strings on their top and bottom (covered in more detail later). Later, *gum* tiles are used to enable power-two gadgets to attach to each other. The gumming process of a label can be seen in Fig. 22a.

Assembling target labels Each round will require the assembly of a subset of labeled assemblies. From round to round, the geometrically-labeled assembly (activator and converter) halves are carried to the next round without change. Thus, in each round, the required labels can be assembled by mixing the appropriate halves in a bin. However, since there can be up to k labels required in each round, using a separate bin for each label exceeds the bin count. To save on bin use, the labels which share a prefix half can be assembled in the same bin (i.e., mix the prefix half into a bin with all required suffix halves for the round). The particular subset of labels required for each round is described later.

Initializing power-two assemblies and target line Initialize power-two assemblies with width 13 and length $2^i - 2$, where i is the smallest integer such that $2^i \geq 4\lceil \log k \rceil + 2$. This length ensures that the segment is long enough to attach labels on both assembly sides. As with prior gadgets, construct power-two assemblies in halves, yielding k distinct labeled power-two assemblies in one bin starting from $\mathcal{O}(\sqrt{k})$ bins (see Fig. 21).

As already done with power-two assemblies, assemble k target lines with distinct labels. The length of some target lines may involve some powers of 2 too short to have labels on their ends. These small values are hard-coded as part of the initial *target line* assemblies well within the bin and stage constraints. After assembly, the initial target line assemblies are placed in a common *gathering bin* that holds partially assembled lines that will grow into completed lines of lengths in $\{n_i\}$.

Growing powers-two assemblies Grow the k power-two assemblies from length $2^i - 2$ to $2^{i+1} - 2$ in $\mathcal{O}(1)$ stages by attaching each assembly to a copy of itself, preserving the matching labels on both sides. First, attach activators to every power-two assembly. Activator assemblies are similar to label assemblies, but are used in the later stages of line building to control the attachment of other assemblies. Second, add gum tiles to enable power-two assembly pairs to attach (see Fig. 22b). This doubling process is repeated until power-two assemblies of length more than n are assembled.

Assembling target lines After each doubling of power-two assembly length, split the resulting bin into two bins; in one, continue the doubling process. The other is used for preparing power-two assemblies for being mixed with partially grown target lines in the gathering bin. Assemble activators for every target line requiring the current round's power-two assembly (this power of two in the line's unique sum), using the $\mathcal{O}(\sqrt{k})$-bin technique previously discussed. Also assemble converter assemblies for all target lines that do *not* require this power-two assembly, with labels of the target line on one side, and the label of another target line that *does* require the power-two assembly on the other side.

Now mix these activators and converters with the power-of-two assemblies; an example of activation is seen in Fig. 23. Add gum tiles to the resulting bin of (activated and converted power-two) assemblies and the gathering bin and mix these bins, causing the desired subset of target lines to grow by the current power of two (see Fig. 24).

The use of converter assemblies is necessary to avoid "pollution" of the gathering bin - the conversion effectively "disposes" of these unneeded power-two assemblies by repurposing them to be used in another target line. In the

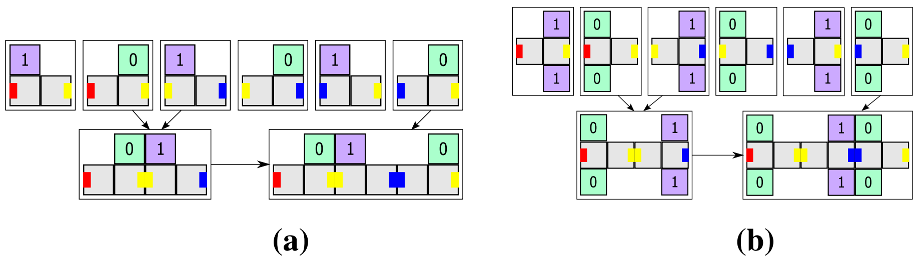


Fig. 20 **a** The 6 three-tile gadgets at top can assemble into a geometric label encoding any x-bit string for $x \in \mathbb{N}$ in $\mathcal{O}(1)$ bins and x stages, including the 3-bit string seen in lower right, by a repeated one-stage mixing of a bin with one tooth into a bin with a growing set of labels. **b** The 6 four-tile gadgets at top can similarly assemble into the activator of any i-bit string

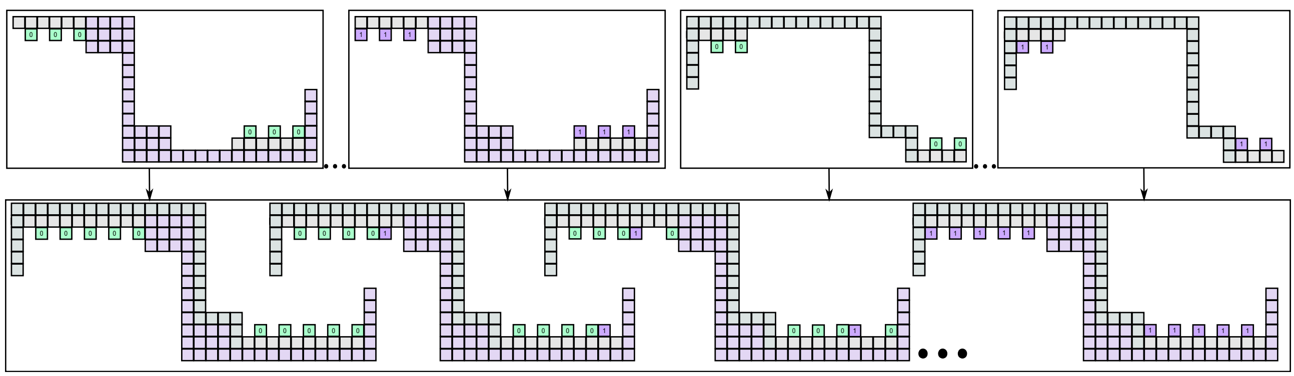


Fig. 21 All $\sqrt{k}$ power-two halves are mixed into a single bin to create all k power-two assemblies

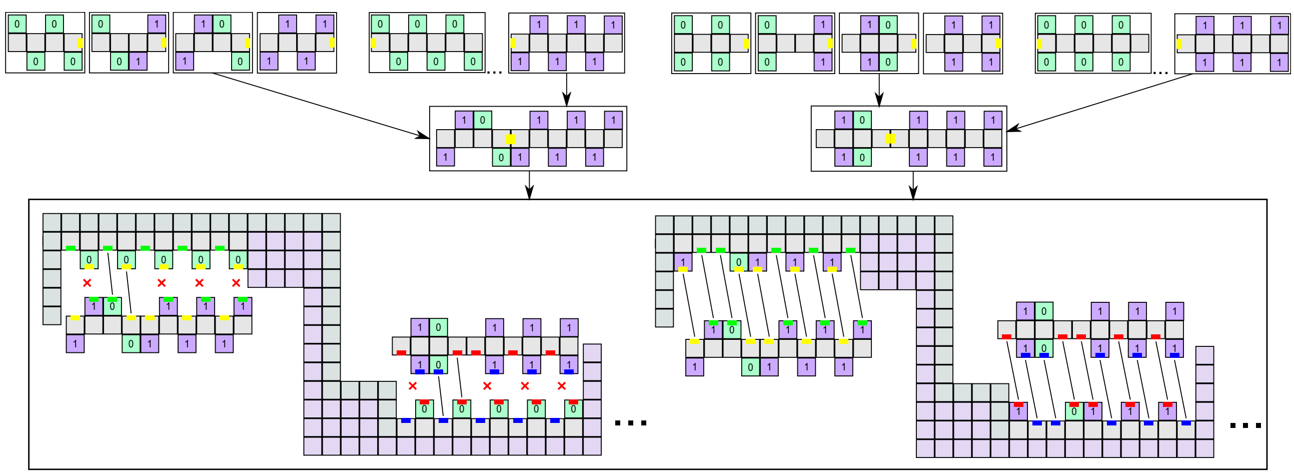


Fig. 22 **a** The $\mathcal{O}(1)$-stage gumming process. The non-descript gum tile (maroon) represents a constant-sized set of tiles which traverse the height of the line. The other tiles (pink) attach to arbitrary length activators. **b** Two gummed copies of the same power-two assembly attaching. (Color figure online)

special case that no power-two assemblies are activated, do not split the initial bin, assemble activator and converters, nor mix into the gathering bin—only carry out the doubling the power-two assembly length.

Once the final round is complete, add a constant-sized set of tiles to the gathering bin to fill in the exposed geometric labels on each constructed line segment, filling it into a full rectangle.

Complexity The gathering bin after each round contains k partially grown target lines, and $\lfloor \log_2 n \rfloor$ rounds are needed to reach the largest necessary bit line (power-of-two) length. Each round uses $\mathcal{O}(1)$ tiles, $\mathcal{O}(\sqrt{k})$ bins, and $\mathcal{O}(1)$ stages ($\mathcal{O}(\log n)$ stages total). Afterwards, $\mathcal{O}(1)$ tiles, bins, and stages are used to "cap-off" the linear assemblies to the proper geometry (rectangles). Also, preparing primitive gadgets uses $\mathcal{O}(1)$ tiles, $\mathcal{O}(\sqrt{k})$ bins, and $\mathcal{O}(\log \sqrt{k})$ stages.

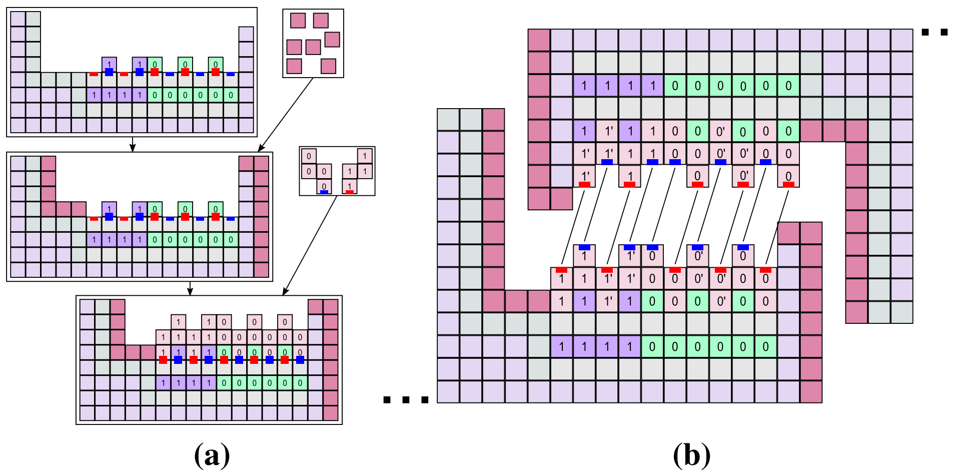


Fig. 23 Selecting specific halves of the activator gadgets (top) and mixing these with the set of all power-two assemblies (bottom) activates only the assembly matching the activator (lower right) and no others (as in lower left)

So in total, $\mathcal{O}(1)$ tiles, $\mathcal{O}(\sqrt{k})$ bins (dominated by assembling sets of activator and converter gadgets), and $\mathcal{O}(\log n)$ stages (dominated by rounds) are used. $\square$

The following lower bound matches this construction and follows directly from Theorem 8.

Corollary 1 *Let $L = \{n_1, \ldots, n_k\} \subseteq \mathbb{N}$ with $n = \max(L)$. For almost all L, any staged self-assembly system with $\mathcal{O}(1)$ tile types and $\mathcal{O}(\sqrt{k})$ bins that assembles $\mathcal{O}(1) \times n_i$ lines for all $n_i \in L$ has $\Omega(\log n)$ stages.*

6 Assembling hefty shapes

The efficient line set assembly result of Theorem 9 can be combined with a technique of Demaine et al. (2015) to assemble general shapes optimally. The technique of Demaine et al. (2015) is to first efficiently create the *backbone* of the given shape, then fill in the backbone of the shape using $\mathcal{O}(1)$ tile types and one stage (see Fig. 25). Given a shape with k edges and vertices, this approach uses $\mathcal{O}(k)$ bins. The bin complexity is reduced to $\mathcal{O}(\sqrt{k})$ by replacing

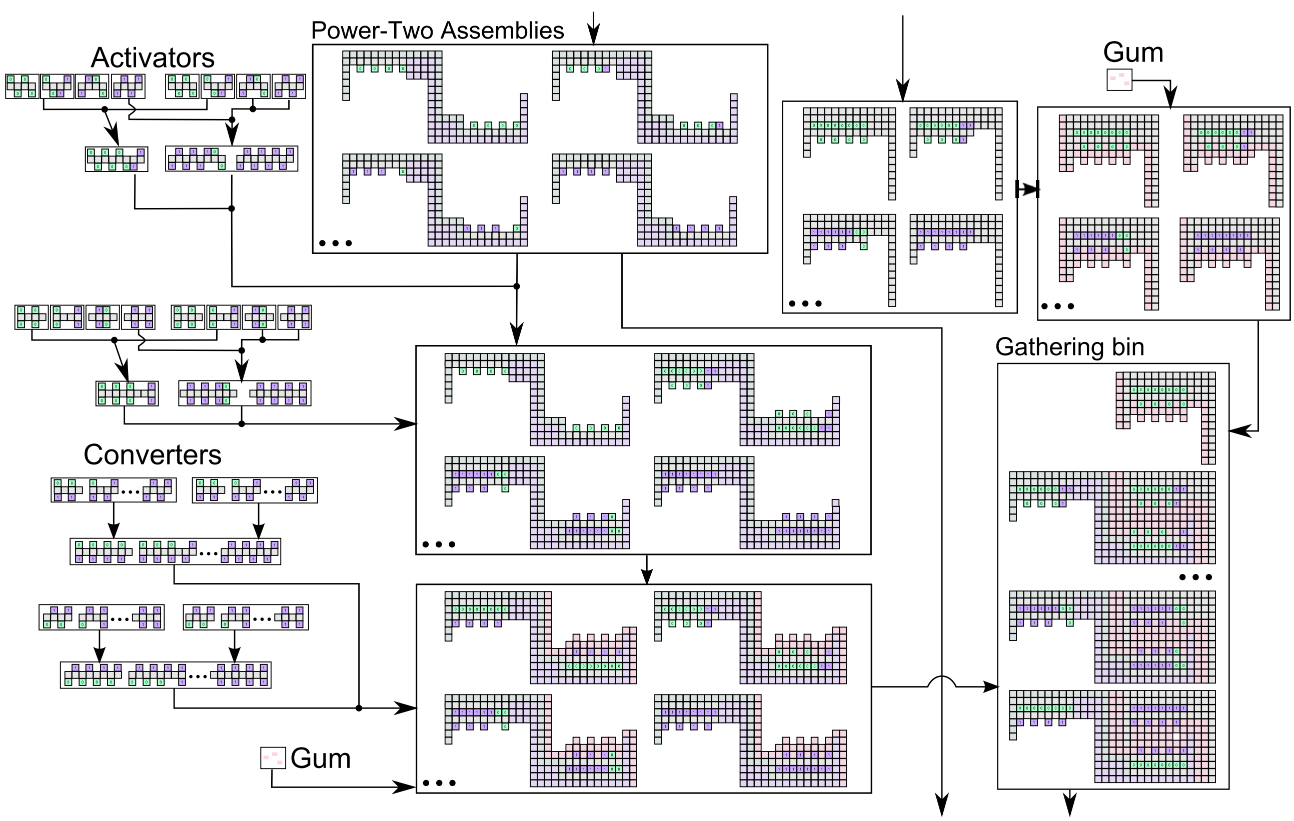


Fig. 24 The portion of the round where power-two assemblies are activated (or converted) to attach to the target lines in the gathering bin

k separate bins, each containing a different edge assembly, with $\mathcal{O}(\sqrt{k})$ bins, each containing many assemblies labeled with geometric teeth, similar to the construction of lines seen in Theorem 9.

Efficient construction of sets of lines as shown in Theorem 9 allows efficient construction of the backbone by assembling the set of lines corresponding to the lengths between the vertices of the backbone of the shape. Constructing these lines requires $\mathcal{O}(\log n)$ stages, since the minimum bounding square of S has edge length n, i.e., the longest line that has to be constructed has length at most n. Geometric teeth are exposed on each line which determine the vertex with which the line will attach. Once the vertices and lines are constructed, they are mixed into one bin to assemble the backbone. Then a $\mathcal{O}(1)$-sized set of filler tiles are added to the backbone which complete the construction of the shape.

Theorem 10 *Let S be a hefty hole-free shape with k vertices and minimum-diameter bounding square of edge length n. There exists a $\tau = 2$ staged system with $\mathcal{O}(\sqrt{k})$ bins, $\mathcal{O}(1)$ tile types, and $\mathcal{O}(\log n)$ stages that uniquely produces S* scaled by a factor $\mathcal{O}(1)$.

Proof To construct the shape S, first consider the shape S scaled by a factor 2, denoted S_2. A point $p \in S_2 \subseteq \mathbb{Z}^2$ is in the *backbone* if one of the 8 neighboring points of p (whose coordinates differ by at most one with p) are not in S_2. The backbone of S_2 has k vertices, each of degree 2. For each of the k vertices, construct a vertex with geometric teeth as shown in Fig. 26a.

To achieve $\mathcal{O}(\sqrt{k})$ bin complexity, vertices are stored in halves and assembled by mixing the appropriate two halves, similar to the mixing used when activating lines in Theorem 9. Strips that connect the vertices of the backbone are assembled using Theorem 9, with the modification that geometric teeth are left exposed on each side of the constructed line, rather than completing the assembly into a rectangle. The lines are long enough to hold geometric teeth due to the assumption that S is hefty. The lines are also modified to have strength-1 south and east facing glues s and e, respectively. Thus mixing the backbone with a bin containing a *filler tile* that exposes glue s to the north and south and glue e to the east and west to fill in the backbone, finalizing the scale 26 version of S.

By Theorem 9, constructing the strips uses $\mathcal{O}(\log n)$ stages, since a strip may not be larger than the bounding square of S, $\mathcal{O}(\sqrt{k})$ bins, since there are $k+1$ strips in the backbone, and $\mathcal{O}(1)$ tile types. Constructing the vertices requires $\mathcal{O}(\sqrt{k})$ bins and $\mathcal{O}(1)$ tile types. The vertices and strips are mixed in one stage and bin, and $\mathcal{O}(1)$ filler tiles are used to fill the backbone in one stage. We consider the backbone of S at scale factor 2, and construct it using our line set construction shown in Theorem 9, which construct lines at a scale factor of 13, for a final assembly of S at scale factor $2 \cdot 13 = 26$. The total complexity of the construction is $\mathcal{O}(\sqrt{k})$ bins, $\mathcal{O}(1)$ tile types, and $\mathcal{O}(\log n)$ stages. □

Theorem 11 *Let S be a hefty shape with k edges and minimum-diameter bounding square of edge length n with $k = \mathcal{O}(n^{2-\varepsilon})$ for some $\varepsilon > 0$. For almost all S, any staged self-assembly system with $\mathcal{O}(1)$ tile types and $\mathcal{O}(\sqrt{k})$ bins that assembles S has $\Omega(\log n)$ stages.*

Proof Consider the set of "comb" shapes with $\Theta(k)$ "teeth" of width $\log(k)$, and length between $\log(k)$ and $n^{\varepsilon/2}$ attached to a "handle" of length $\Theta(k \log k)$ and separated by distance at least $\log(k)$ along the handle. There are $n^{\Theta(k)}$ such shapes, and for some choice of constants, they have k edges and minimum-diameter bounding squares with edge length n. So almost all shapes in this set, and thus in the superset of hefty shapes defined in the theorem statement, require $\log(n^{\Omega(k)}) = \Omega(k \log n)$ bits to specify. By Lemma 4, any staged system with $\mathcal{O}(1)$ tile types and $\mathcal{O}(\sqrt{k})$ bins requires $\mathcal{O}(sk)$ bits to specify. So almost all shapes in the set specified by the theorem statement can only be assembled by such a staged system with $\Omega(\frac{k \log n}{k}) = \Omega(\log n)$ stages. □

7 Conclusion and future work

In this work, we established the first general trade-off bounds for benchmark problems in staged self-assembly, specifically for thickness-one ($1 \times n$) and thin ($\mathcal{O}(1) \times n$)

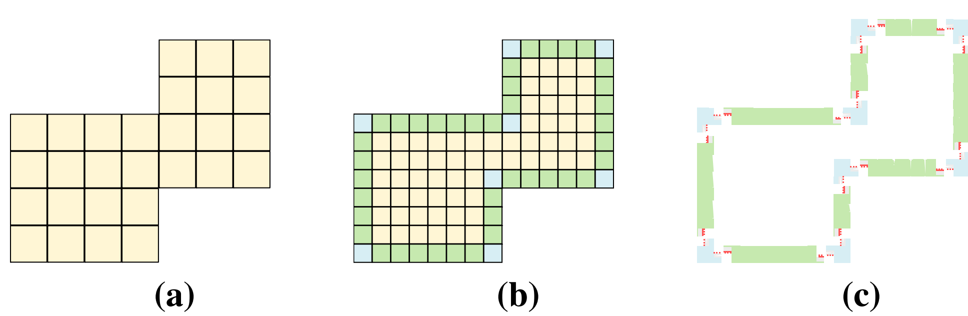


Fig. 25 **a** A hefty hole-free shape to be constructed. **b** The shape scaled by factor 2 with backbone (green) and vertices (blue). **c** The decomposition of the backbone into vertices and lines. (Color figure online)

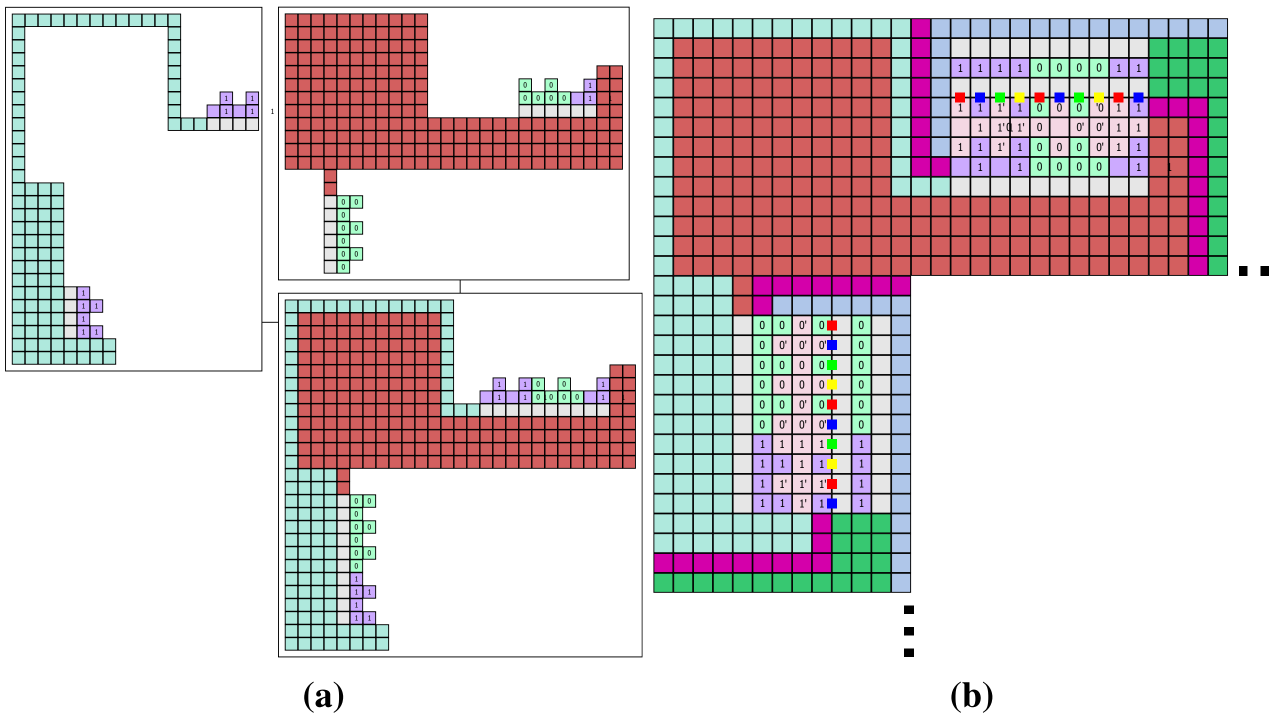


Fig. 26 **a** The mixing of two vertices to construct a vertex with geometric teeth. By storing the vertex in halves, we can achieve $\mathcal{O}(\sqrt{k})$ bin complexity. The teeth on the vertex decide which length line segment attaches to that vertex. **b** A vertex having attached the appropriate line segments

linear assemblies. We also introduced two approaches to simultaneously assembling large sets of linear assemblies with distinct lengths, and applied them to assembling the general class of "hefty" shapes, reducing the number of bins used by prior work and answering a question asked by Demaine et al. (2015). However, there remain numerous open problems and directions:

- Can the additive gap of $\mathcal{O}(\frac{\log \log b}{\log t})$ between Theorems 4 and 6 (and their flexible glue equivalents) be reduced? This additive gap is incurred by a specific subroutine in our algorithms, and focusing on improving this subroutine may be fruitful.
- In any realistic implementation for building $\mathcal{O}(1) \times n$ linear assemblies, the exponential garbage produced in our construction would be a problem. Can the same results be achieved without building all combinations and throwing away the incorrect ones? Can a construction work that limits the garbage to be polynomial in the size of the final assembly?
- Can similar results be obtained in weakened versions of the model obtained by reducing the temperature to 1, yielding a "fully connected" final assembly, or using only "planar" assembly or "size-separable" mixings? See Demaine et al. (2008, 2015), Winslow (2016) for definitions and partial results.
- What is the complexity of prediction and optimization problems in staged assembly? For example, the problem of predicting whether a given staged system uniquely assembles a target assembly has been shown to be $\mathsf{coNP}^{\mathsf{NP}}$-hard and in PSPACE Schweller et al. (2017) as well as related optimization problems Demaine et al. (2013), Winslow (2015), Schweller et al. (2017), but no substantial collection of such results exists as it does for the 2HAM Cannon et al. (2013), Schweller et al. (2017).
- Substantial recent work in tile assembly has focused on comparing models using simulation definitions that capture not only computational, but also geometric behavior Woods (2015). No such simulation results have related the staged model to other two-handed models, or to itself with varying parameters. Is it possible that unlike the 2HAM Demaine et al. (2016), the staged model is intrinsically universal for all temperatures?

References

Adleman L, Cheng Q, Goel A, Huang MD, Wasserman H (2001) Linear self-assemblies: equilibria, entropy and convergence rates. In: 6th international conference on difference equations and applications

Adleman LM, Cheng Q, Goel A, Huang MDA, Kempe D, de Espanés PM, Rothemund PWK (2009) Combinatorial optimization problems in self-assembly. SIAM J Comput 38(6):2356–2381

Barish RD, Schulman R, Rothemund PWK, Winfree E (2009) An information-bearing seed for nucleating algorithmic self-assembly. Proc Nat Acad Sci 106(15):6054–6059

Cannon S, Demaine ED, Demaine ML, Eisenstat S, Patitz MJ, Schweller RT, Summers SM, Winslow A (2013) Two hands are better than one (up to constant factors): Self-assembly in the 2HAM vs. aTAM. In: STACS 2013, LIPIcs, vol 20. p 172–184. Schloss Dagstuhl

Chalk C, Martinez E, Schweller R, Vega L, Winslow A, Wylie T (2018) Optimal staged self-assembly of general shapes. Algorithmica 80(4):1383–1409. https://doi.org/10.1007/s00453-017-0318-0

Chandran H, Gopalkrishnan N, Reif J (2012) Tile complexity of linear assemblies. SIAM J Comput 41(4):1051–1073

Chen HL, Doty D (2012) Parallelism and time in hierarchical self-assembly. In: 23rd annual ACM-SIAM symposium on discrete algorithms (SODA), p 1163–1182. SIAM

Cheng Q, Aggarwal G, Goldwasser MH, Kao MY, Schweller RT, de Espanés PM (2005) Complexities for generalized models of self-assembly. SIAM J Comput 34:1493–1515

Demaine ED, Demaine ML, Fekete SP, Ishaque M, Rafalin E, Schweller RT, Souvaine DL (2008) Staged self-assembly: nanomanufacture of arbitrary shapes with $O(1)$ glues. Nat Comput 7(3):347–370

Demaine ED, Eisenstat S, Ishaque M, Winslow A (2013) One-dimensional staged self-assembly. Nat Comput 12(2):247–258

Demaine ED, Fekete SP, Scheffer C, Schmidt A (2015) New geometric algorithms for fully connected staged self-assembly. In: Proceedings of the 21st international conference on DNA computing and molecular programming, LNCS, vol 9211. Springer, p 104–116

Demaine ED, Patitz MJ, Rogers TA, Schweller RT, Summers SM, Woods D (2016) The two-handed tile assembly model is not intrinsically universal. Algorithmica 74(2):812–850

Evans C (2014) Crystals that count! Physical principles and experimental investigations of dna tile self-assembly. Ph.D. thesis, Caltech

Rothemund PWK, Winfree E (2000) The program-size complexity of self-assembled squares (extended abstract). In: Proceedings of the 32nd ACM symposium on theory of computing, STOC'00, p 459–468

Schulman R, Winfree E (2007) Synthesis of crystals with a programmable kinetic barrier to nucleation. Proc Nat Acad Sci 104(39):15236–15241

Schweller R, Winslow A, Wylie T (2017) Complexities for high-temperature two-handed tile self-assembly. In: Proceedings of the 23rd international conference on DNA computing and molecular programming

Schweller R, Winslow A, Wylie T (2017) Verification in staged tile self-assembly. In: 16th international conference unconventional computation and natural computation, p 98–112

Seeman NC (1982) Nucleic-acid junctions and lattices. J Theor Biol 99:237–247

Tikhomirov G, Petersen P, Qian L (2017) Fractal assembly of micrometre-scale dna origami arrays with arbitrary patterns. Nature 552(7683):67

Winfree E (1998) Algorithmic self-assembly of DNA. Ph.D. thesis, Caltech

Winslow A (2015) Staged self-assembly and polyomino context-free grammars. Nat Comput 14(2):293–302

Winslow A (2016) Size-separable tile self-assembly: a tight bound for temperature-1 mismatch-free systems. Nat Comput 15(1):143–151

Woods D (2015) Intrinsic universality and the computational power of self-assembly. Philos Trans R Soc A. https://doi.org/10.1098/rsta.2014.0214