Randomized Self-Assembly for Approximate Shapes

Ming-Yang Kao[⋆] and Robert Schweller[⋆⋆]

Abstract. In this paper we design tile self-assembly systems which assemble arbitrarily close approximations to target squares with arbitrarily high probability. This is in contrast to previous work which has only considered deterministic assemblies of a single shape. Our technique takes advantage of the ability to assign tile concentrations to each tile type of a self-assembly system. Such an assignment yields a probability distribution over the set of possible assembled shapes. We show that by considering the assembly of close approximations to target shapes with high probability, as opposed to exact deterministic assembly, we are able to achieve significant reductions in tile complexity. In fact, we restrict ourselves to constant sized tile systems, encoding all information about the target shape into the tile concentration assignment. In practice, this offers a potentially useful tradeoff, as large libraries of particles may be infeasible or require substantial effort to create, while the replication of existing particles to adjust relative concentration may be much easier. To illustrate our technique we focus on the assembly of $n \times n$ squares, a special case class of shapes whose study has proven fruitful in the development of new self-assembly systems.

Key words: Self-Assembly, Randomized Algorithms

1 Introduction

Self-assembly is the process by which simple objects autonomously assemble into complexes. This phenomenon is common in nature and is the mechanism behind many natural occurrences such as crystal growth. Current research is particularly interested in understanding and harnessing the power of self-assembly for the purpose of massive fabrication of nanoscale devices such as computer circuits. In particular, researchers have identified DNA molecules as a promising medium in which to design controlled self-assembly systems for nanomanufacturing and biologically based computing.

The leading theoretical model for self-assembly is the tile assembly model introduced by Winfree [15]. The tile assembly model extends the theory of Wang tilings of the plane [14] by adding a natural mechanism for growth. Informally, particles of a self-assembly system are modeled by four-sided Wang tiles, each with a type of glue assigned to each side. The tiles float about in the plane and stick together when they bump if the affinity between touching glues is strong enough. In this way, the particles or tiles of the system self-assemble into a complex pattern or shape. The goal is then as follows: Given a target shape or pattern, design a system of tiles that will assemble into the target. The quality or efficiency of the system is then measured by how few distinct tile types are used. This measurement is motivated by the fact that each distinct type of tile must be manufactured if the system is to be implemented.

The tile model of self-assembly is primarily motivated by a DNA based implementation. Double and triple crossover DNA molecules have been designed that can act as four-sided building blocks (tiles) for DNA self-assembly [7, 9]. Experimental work has been done to show the effectiveness of using these tiles to assemble DNA crystals and perform DNA computation [10, 11, 16, 17].

Traditional work in this field has taken the approach of encoding information about the target shape into the tile types of the system [12, 1–3, 13]. In particular, Rothemund et al. [12] and Adleman et al.[1] show how the assembly of $n \times n$ squares can be assembled using $\Theta(\frac{\log n}{\log \log n})$ distinct tile types. However, the design of large sets of distinct tile types can be problematic. Many mediums of self-assembly may have small practical limitations on the number of glues or tiles that can be manufactured. Even in the most promising scenario of DNA self-assembly in which glues and tiles are encoded with long strands of DNA, there is an associated computational complexity with the design of large sets of DNA glues, as well as the likely need to redesign DNA tile structure for increasingly large tile sets.

[⋆] Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA. Email: kao@northwestern.edu.

[⋆⋆] Department of Computer Science, University of Texas-Pan American, Edinburg, TX 78539, USA. Email: schwellerr@cs.panam.edu.

In contrast, recent work has be done examining the possibility of encoding the complexity of the target shape outside of the particles in the system entirely. In [8], we showed that there exists a constant sized tile set that can effectively be programmed to assemble any $n \times n$ square with affecting a short sequence of temperatures in the system. Further, Demaine et. al. [6] showed how constant size tile sets can build arbitrary shapes by mixing intermediate self-assembly batches together in a sequence of stages. However, with these techniques the complexity of the target shape is contained within a sequence of laboratory steps, rather than within the system of particles itself.

In this paper we take a new approach. We do encode the complexity of the target shape within the particles of the system itself. But, we avoid the problems of encoding complexity into the distinct tile types of the system. Rather, we encode the complexity into the relative concentrations of tiles in the system. While completely encoding the complexity of the target shape into the particles of the system, we use only a $O(1)$ size set of distinct tiles. In many instances, in particular in the case of DNA, design of distinct, new particles or tiles is much more difficult than creating a large number of copies of a given particle or tile type. Thus, the implementation of a set of relative tile concentrations for a small, universal set of tile types is potentially a more practical approach than the implementation of a completely new set of tiles. Further, for systems based on an implementation other than DNA, there are likely even more difficulties for the design of large distinct tile systems. In particular, the design of different glues based on protein-protein interactions is very limited, making a scheme with a fixed number of required glue types even more desirable.

1.1 Our Technique

Our motivating example in this paper is the assembly of $n \times n$ squares. The key to the assembly of an $n \times n$ square is the ability to create a supertile that encodes an arbitrary length $\log n$ binary string [12]. At a high level, our technique is to design a fixed size tile set that is capable of encoding, with high precision, such a string into a supertile as a function of tile concentration assignment. In particular, we design a tile set that assembles a large two dimensional array of tiles whose inner pattern of tiles constitutes a sampling of tile types. The relative number of occurrences of certain tile types within this structure provides sampled information about the relative tile concentration assignment of the tile set. By combining this sampling array with tiles capable of counting and performing arithmetic, this sampled information can be extracted into the form of a binary string displayed on the surface of the assembled supertile. Once the estimated binary string is displayed it is a straightforward extension to utilize binary counters, as in [12], to complete an $n \times n$ square where the n is specified by the estimated binary string.

To make this technique work, we need to accomplish two contradictory objectives: First, the dimensions of the sampling array supertile must be small so that they do not go beyond the width or height of the target $n \times n$ square. Second, the area of the sampling array must be large enough so that the sample obtained yields a provable accuracy guarantee from Chernoff bounds.

1.2 Our Results

Our results are summarized as follows. First, we consider the (ϵ, δ)-approximate assembly of $n \times n$ squares. A system is said to achieve an (ϵ, δ)-approximate assembly of an $n \times n$ square if the system will assemble an $n' \times n'$ square with probability at least $1 - \delta$ such that $(1 - \epsilon)n \leq n' \leq (1 + \epsilon)n$. We show that for any ϵ and δ and sufficiently large n (as a function of ϵ and δ), there exists a tile set of size $O(1)$ that achieves an (ϵ, δ)-approximate assembly of an $n \times n$ square. In contrast, the best result for the exact, deterministic assembly of $n \times n$ squares requires $O(\frac{\log n}{\log \log n})$ distinct tiles.

1.3 Related Work

Our results build on a recent technique proposed by Becker et al. [5] showing that if we wish to assemble several shapes, it is possible to reduce the total complexity needed by assembling the shapes together using a combined tile system rather than by assembling the shapes individually using separate tile systems. Specifically, they provide $O(1)$ size tile sets that build squares and rectangles of expected dimension n, where the n is specified by tile concentrations. While this is an initial step towards concentration based control, their techniques yield a large variance and thus do not provide the precise control achieved in our work.

Previous research has considered the use of tile concentration assignments for the purpose optimizing assembly time [1]. However, they do not consider the effect concentration assignment has on the final assembled shape.

1.4 Paper Layout

The remainder of this paper is organized as follows. In Section 2 we introduce the tile assembly model. In Section 3 we discuss a preliminary, low precision technique for controlling the expected size of a target shape. In Section 4 we introduce a randomized sampling technique to assemble high precision approximations of target binary numbers. In Section 5 we apply the randomized sampling technique to the assembly of (ϵ, δ)-approximate squares to obtain our main result. In Section 6 we conclude with a discussion of future research directions.

2 Basics

2.1 Definitions

To describe the tile self-assembly model, we make the following definitions. A tile t in the model is a four sided Wang tile denoted by the ordered quadruple $(\text{north}(t), \text{east}(t), \text{south}(t), \text{west}(t))$. The entries of the quadruples are glue types taken from an alphabet Σ representing the north, east, south, and west edges of the Wang tile, respectively. A *tile system* is an ordered quadruple $\langle T, s, G, \tau, P\rangle$ where T is a set of tiles called the *tileset* of the system, τ is a positive integer called the *temperature* of the system, $s \in T$ is a single tile called the *seed* tile, G is a function from Σ^2 to $\{0, 1, \ldots, \tau\}$ called the *glue function* of the system, and P is a function denoting a probability distribution over the set of tiles in T representing the relative concentrations of each tile type. It is assumed that $G(x, y) = G(y, x)$, and there exists a null in Σ such that $\forall x \in \Sigma$, $G(\texttt{null}, \texttt{x}) = \texttt{0}$. In this paper we assume the glue function is such that $G(x, y) = 0$ when $x \neq y$ and denote $G(x, x)$ by $G(x)$ (see [3, 4] for the effect of removing this restriction). $|T|$ is referred to as the *tile complexity* of the system. In this paper we also only consider temperature $\tau = 2$.

Define a *configuration* to be a mapping from $\mathbb{Z}^2$ to $T \bigcup \{\texttt{empty}\}$, where empty is a special tile that has the null glue on each of its four edges. The *shape* of a configuration is defined as the set of positions (i, j) that do not map to the empty tile. For a configuration C, a tile $t \in T$ is said to be *attachable* at the position (i, j) if $C(i, j) = \texttt{empty}$ and $G(\text{north}(t), \text{south}(C(i, j+1))) + G(\text{east}(t), \text{west}(C(i+1, j))) + G(\text{south}(t), \text{north}(C(i, j-1))) + G(\text{west}(t), \text{east}(C(i-1, j))) \geq \tau$. For configurations C and C' such that $C(x, y) = \texttt{empty}$, $C'(i, j) = C(i, j)$ for all $(i, j) \neq (x, y)$, and $C'(x, y) = t$ for some $t \in T$, define the act of *attaching* tile t to C at position (x, y) as the transformation from configuration C to C'. For a given tile system $\mathbf{T}$, if a supertile B can be obtained from a supertile A by the addition of a single tile we write $A \rightarrow_T B$. Further, we denote $A \rightarrow_T$ as the set of all B such that $A \rightarrow_T B$ and $\rightarrow_T^*$ as the transitive closure of $\rightarrow_T$.

Define the *adjacency graph* of a configuration C as follows. Let the set of vertices be the set of coordinates (i, j) such that $C(i, j)$ is not empty. Let there be an edge between vertices (x_1, y_1) and (x_2, y_2) iff $|x_1-x_2|+|y_1-y_2| = 1$. We refer to a configuration whose adjacency graph is finite and connected as a *supertile*. For a supertile S, denote the number of non-empty positions (tiles) in the supertile by $\text{size}(S)$. We also note that each tile $t \in T$ can be thought of as denoting the unique supertile that maps position $(0, 0)$ to t and all other positions to empty. Throughout this paper we will informally refer to tiles as being supertiles.

2.2 The Assembly Process

Deterministic Assembly Assembly takes place by *growing* a supertile starting with tile s at position $(0, 0)$. Any $t \in T$ that is attachable at some position (i, j) may attach and thus increase the size of the supertile. For a given tile system, any supertile that can be obtained by starting with the seed and attaching arbitrary attachable tiles is said to be *produced.* If this process comes to a point at which no tiles in T can be added, the resultant supertile is said to be *terminally* produced. For a given shape Υ, a tile system Γ *uniquely produces* shape Υ if for each produced supertile A, there exists some terminally produced supertile A' of shape Υ such that $A \rightarrow_T^* A'$. That is, each produced supertile can be grown into a supertile of shape Υ. This definition of unique assembly is introduced in [3] and differs slightly from previous work [12, 1, 1] in that we do not require that a unique supertile be terminally assembled. The *tile complexity* of a shape Υ is the minimum tile set size required to uniquely assemble Υ.

Probabilistic Assembly In addition to considering tile systems that uniquely assemble a given shape, we can consider systems that can potentially build multiple shapes, but will build one of a desired class of shapes with high probability. To study this model we can think of the assembly process as a Markov chain where each producible supertile is a state and transitions occur with non-zero probability from supertile A to each $B \in A \rightarrow_T$. For each $B \in A \rightarrow_T$, let t_B denote the tile added to A to get B. The transition probability from A to B is defined to be

$$\text{TRANS}(A, B) = \frac{P(t_B)}{\sum_{C \in A \rightarrow_T} P(t_C)}.$$

The probability that a tile system T terminally assembles a supertile A is thus defined to be the probability that the Markov chain ends in state A. Further, the probability that a system terminally assembles a shape Υ is the probability the chain ends in a supertile state of shape Υ.

3 Low Precision Technique (Line Approximation)

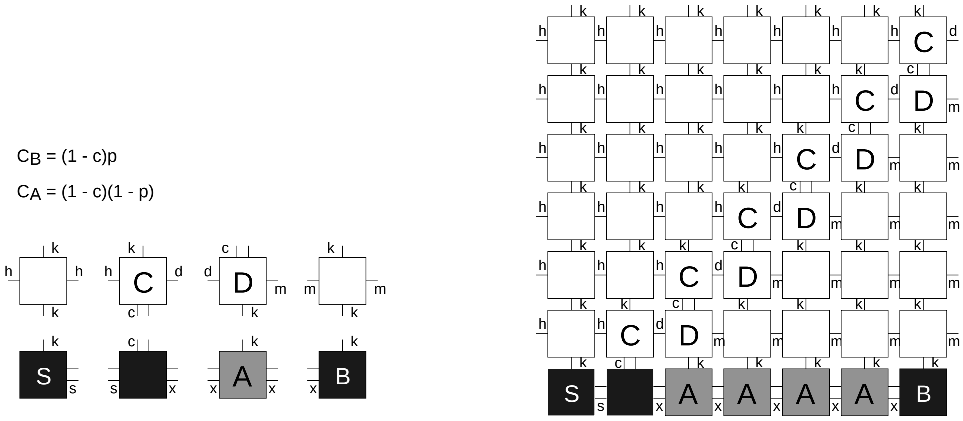

Fig. 1. With S as the seed tile, these tiles can assemble into all $n \times n$ squares with $n \geq 3$. The concentrations of tile types A and B are denoted as C_A and C_B. With $p = \frac{1}{n}$ and c being the sum of all tile concentrations other than those for A and B, the expected dimension of the assembled square is n.

Becker et al. [5] first considered the assignment of tile type concentrations for the control of assembled shapes. They consider a tileset of 5 tile types whose set of terminally produced supertiles is the set of all squares. Further, they show how for any given n, a corresponding tile concentration assignment ensures that the expected dimension of the assembled square is n. A modified version of this tile set is described in Figure 1.

The basic method of this tile system is the assembly of the line consisting of the seed, tile A, and the final tile B. As the line grows from left to right, each position can potentially be filled with an A tile, in which case the line continues to grow, or a B tile, in which case the line stops growing. If the probability of placing the B tile is p and placing the A tile is $1 - p$, then the length of the assembled line follows a geometric distribution and has expected length $\frac{1}{p}$. An expected dimension of n can thus be achieved by setting concentrations so that $p = \frac{1}{n}$.

With this method, a square is assembled whose dimension is determined by a single line of tiles whose length is a random variable with a geometric distribution. The problem with this technique is that the length of the line has a high variance. Our improved technique will create a different supertile for encoding the target dimension n. This supertile will provided an estimate for the target dimension that follows a binomial distribution instead of a geometric distribution. With this technique it is then possible to apply Chernoff bounds to achieve much more precise results.

4 The Basic Idea (Sampling Approximation)

Our goal is to assemble a supertile that encodes an x-bit binary string b. Let n be the value of the string when interpreted as a binary number. We will say that our scheme builds an (ϵ, δ)-approximate version of n if the supertile assembled encodes a value of n', $(1-\epsilon)n \leq n' \leq (1+\epsilon)n$, with probability at least $1-\delta$. For the low precision technique, the encoding of the target n is simply the length of the assembled a, b line. Note that this does not yield an (ϵ, δ)-approximate scheme for small ϵ and δ.

Our technique combines the line approximation with tiles capable of performing binary counting and binary division. First consider the line technique modified in Figure 2 so that there are two types of A tile, one of them red. Conditional on the event that one of the two types of A tiles is placed, we can control with tile concentrations that the probability of placing a red tile be some desired value q.

Now consider the random variable R denoting the sum of all red tiles placed before the final B tile is placed. Let L denote the length of the line. R then has a binomial distribution with $\mu = qL$. The goal is then to add to this construction (1) tiles capable of computing R, (2) tiles for computing L, and (3) tiles capable of performing division, in particular, tiles for computing $\frac{R}{L}$. With such tiles, we can assemble a string of tiles that encode the random variable $\frac{L}{R}$. By setting $q = \frac{1}{n}$, this random variable has expectation $\frac{L}{qL} = n$. And since R has a binomial distribution, we can apply Chernoff bounds to bound the tail probabilities of this variable when its expectation qL is high enough. While we cannot control L with high precision, we can ensure that with high probability it is sufficiently large, making the expectation of R large as well. In more detail, the tiles for the construction are as follows.

Sampling Line The sampling line consists of the seed tile S and the three tiles in Figure 2. The expected length of the line will be $\frac{1}{p}$, and the expected ratio of the length to the number of red tiles will be n. Further, with high probability (at least $1-\delta$) the sampling line will be long enough to guarantee that this fraction is within an ϵ factor of the target n.

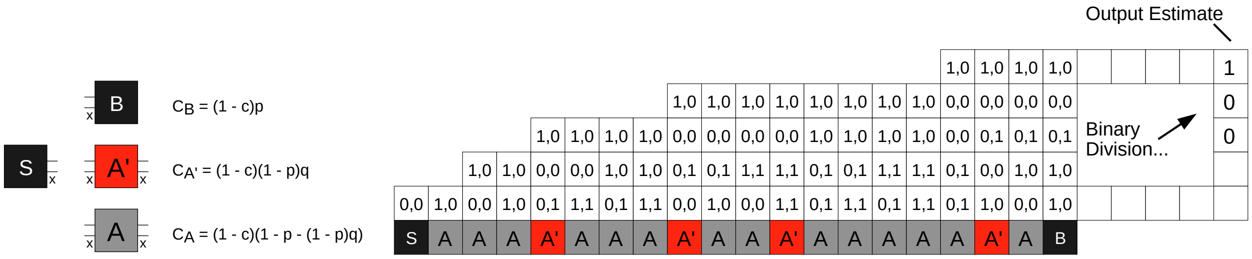

Fig. 2. With S as the seed tile, these tiles form the sampling line. The concentrations of each tile are noted as C_A, $C_{A'}$, and C_B. Here $q = \frac{1}{n}$ and p is any value at most $1-(1-\delta)^{\frac{\epsilon^2}{(1+\epsilon)^2 3n \ln \frac{3}{\delta}}}$. The value c denotes the total concentration from all other tiles within the tile system. These tiles create a line, sampling a red tile with a density of $1/n$. A binary counter sums up the total length, as well as the total number of red tiles. These two values are then divided to yield an estimate for n.

Double Counter Tiles capable of counting in binary are known [12]. Straightforward modifications are possible to permit the simultaneous counting of multiple values as shown in Figure 2. Here each column of tiles represents two counts, one for the total length of the sampling line covered from the seed up until the current column, and the other the number of red tiles covered. These numbers are represented in binary with the duple label of the tile in the i^{th} row representing the ith bit of the length counter as the first coordinate and the i^{th} bit of the red tile counter as the second. Exact details for this construction are omitted in this extended abstract.

Binary Division Work has been done to show how to perform arithmetic with self-assembly [15]. We can apply a modified set of division tiles to compute $\frac{L}{R}$ from the values L and R encoded in the double counter. Details are omitted in this extended abstract.

Theorem 1. *For any given $\epsilon, \delta \leq 0$ and positive integer n, the sampling approximation tile system creates an estimate whose value n' is such that $(1-\epsilon)n \leq n' \leq (1+\epsilon)n$ with probability at least $1-\delta$.*

Proof. Let L be the random variable denoting the length of the sampling line and the R be the random variable denoting the number of red tiles in the line. From the assigned tile concentrations, L is has a geometric distribution with mean $\frac{1}{p}$ and R has a binomial distribution with mean qL. By applying Chernoff bounds we get that

$$P[R > (1 + \frac{\epsilon}{1+\epsilon})qL] < \exp(-qL\epsilon^2/3(1+\epsilon)^2) \tag{1}$$

and

$$P[R < (1 - \frac{\epsilon}{1+\epsilon})qL] < \exp(-qL\epsilon^2/2(1+\epsilon)^2). \tag{2}$$

Our goal is now to ensure that L is large enough so that the above equations are bounded by $(1-\frac{\delta}{3})$. Since L has a geometric distribution, for any positive integer x we have that $P[L > x] = (1-p)^x$. Thus, for $x = \frac{(1+\epsilon)^2 3n \log\frac{3}{\delta}}{\epsilon^2}$ and $p = 1 - (1-\frac{\delta}{3})^{\frac{\epsilon^2}{(1+\epsilon)^2 3n \log\frac{3}{\delta}}}$, we get that $(1-p)^x = 1 - \frac{\delta}{3}$. Thus,

$$P[L > \frac{(1+\epsilon)^2 3n \log\frac{3}{\delta}}{\epsilon^2}] = 1 - \frac{\delta}{3}. \tag{3}$$

Further, by plugging x into the right hand sides of equations (1) and (2), we get that $(1-\frac{\epsilon}{1+\epsilon})qL \leq R \leq (1+\frac{\epsilon}{1+\epsilon})qL$ with probability at least $1-\frac{2\delta}{3}$ when $L \geq x$. Combining this with the probability bound from equation (3) yields the following

$$(1 - \frac{\epsilon}{1+\epsilon})qL \leq R \leq (1 + \frac{\epsilon}{1+\epsilon})qL \text{ with probability at least } 1-\delta. \tag{4}$$

Finally, now consider the tile system's estimate of $\frac{L}{R}$. From equation (4) we get that with probability at least $1-\delta$

$$\frac{L}{(1+\frac{\epsilon}{1+\epsilon})qL} \leq \frac{L}{R} \leq \frac{L}{(1-\frac{\epsilon}{1+\epsilon})qL} \tag{5}$$

$$\Rightarrow \frac{(1+\epsilon)n}{1+2\epsilon} \leq \frac{L}{R} \leq (1+\epsilon)n \tag{6}$$

$$\Rightarrow (1-\epsilon)n \leq \frac{L}{R} \leq (1+\epsilon)n. \tag{7}$$

This completes the proof of Theorem 1. □

5 $n \times n$ Squares

In this section we apply the basic technique of sampling approximation to the assembly of approximate $n \times n$ squares. As shown in [12], there exists a general set of *square building* tiles of constant size that, given a supertile that encodes a length $\log n$ binary string n' (the string encoded is not exactly n but uniquely identifies it) will uniquely assemble into an $n \times n$ square. The key is then to efficiently build such a supertile. This can be done trivially with $\log n$ distinct tile types, while a more efficient method yields $O(\frac{\log}{\log\log n})$ tile types [1], which is optimal for almost all n. We instead will use the sampling approximation to achieve the result with only $O(1)$ total tiles.

However, there is a problem with directly using the line approximation from Section 4. To approximate a value n' with small values of ϵ and δ, the length of the sample line must be many times larger than n'. As n' can be almost as large as n, the length of the estimation line will far exceed the width n of the square and will thus fail to build the square.

However, we can get around this problem by taking advantage of the extra dimension of the square. That is, while the width of the square is n, there is actually n^2 space within the bounds of the square. It is plausible then that an approximation line of length many times n, broken up into pieces of length at most n, could fit within the boundary of an $n \times n$ square. We do exactly this with what we call the *approximation frame.*

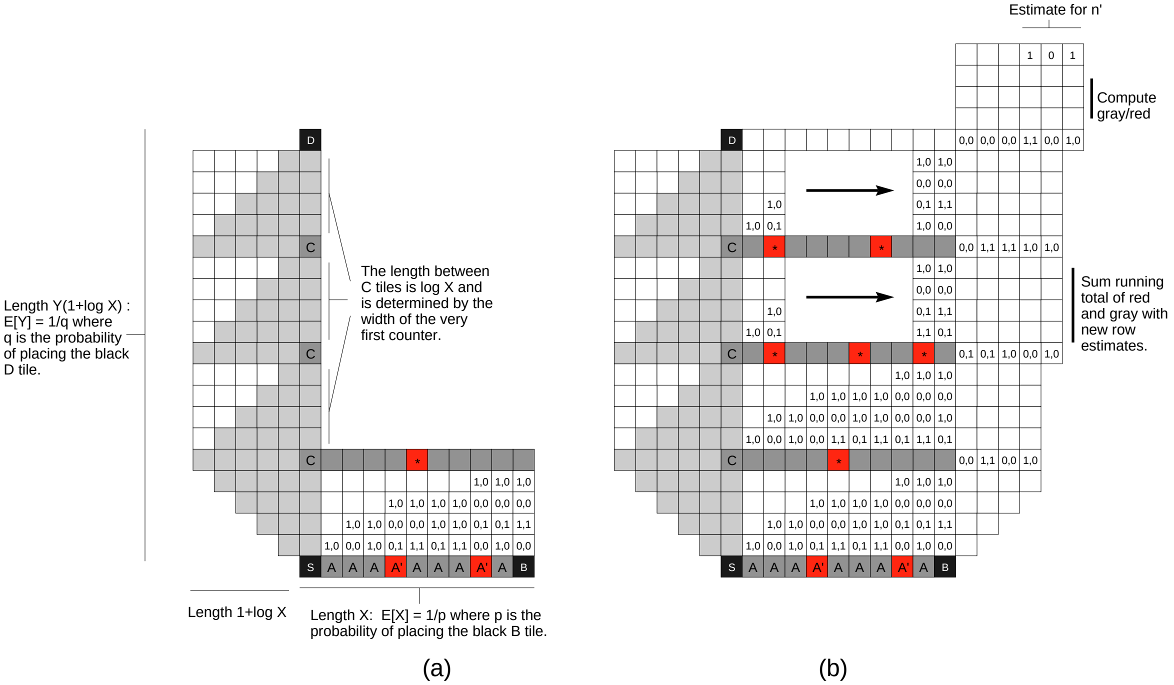

Fig. 3. This is a high level schemata for how the estimation frame can be used to build $n \times n$ squares.

5.1 Approximation Frame

The basic idea for the approximation frame is to use two separate line approximations, one line growing east of the seed and another growing north. Then, for each of the tiles in the vertical line, create a sampling line that grows east up to the length of the initial horizontal line approximation. By further providing sufficient space between each tile in the vertical approximation line, we can use counter tiles to sum the number of red tiles in each approximation line. Each individual sum for each sampling line can then be summed to gain a total number of red tiles, as well as the total length, to compute an estimate for the target n'.

The key to this technique is two fold. First, we need to be able to say with high probability that both dimensions of the frame are small enough so that they do not exceed the width n of the target square. Second, we need that with high probability the total length of all the approximation lines, which is the length of the initial horizontal approximation line multiplied by the length of the initial vertical approximation line, is sufficiently large to provide an accurate estimate of the target n'. In this section we show that for any given ϵ and δ this is possible for large enough n'.

In Figure 3 the high level structure of the approximation frame is described. Starting from the seed tile, a sampling line grows east with length X, which has expectation $\frac{1}{p_x}$, and the expected ratio of red tiles to the total length equal to $q = \frac{1}{n'}$. A double counter computes both the length and the number of red tiles, as in the basic sampling line. However, in this case the final value of the counter seeds a new row of tiles that grows back in the direction of the seed, again sampling a red tile with probability q. This return row places the black S' tile which seeds a new approximation line, this time growing north. However, each placement of a C tile in this approximation line is buffered by a distance of $\log X$ tiles, i.e., the length of the counter that computes the length X. It is straightforward to maintain this buffer by using the initial length between S and S' as a *yardstick*. A diagonal growth of tiles, depicted as the grey tiles in the figure, can translate a vertical length into a horizontal length, and vice versa. With these tiles the original length between S and S' can be placed before each C tile until the final D is placed. Let Y denote the length of this approximation line when only counting the C and D tiles. Thus, the total length is $Y \log X$ with expectation $\frac{\log X}{p_y}$ if p_y is the probability of placing D instead of C.

For each C, a new sampling line is constructed, with the distinction that the length is deterministically equal to the initial random length X. For each of these rows a double counter is used to calculate the length and number of red tiles. Since the maximum bits of the double counter is the same as with the initial count, we are guaranteed to have enough room for the counter to complete.

Finally, tiles for summation are used along the east side of the frame to calculate the total sum of red tiles as well as the total length of the sampling lines (which is XY). A final division is performed to compute the ratio of length to red tiles for the estimate.

Dimensions of Frame As initially mentioned in this section, one potential problem with this construction is that the estimation frame can grow to exceed n in one of its dimensions, making the assembly of any square of dimension at most n impossible. The dimensions of the frame are as follows.

Horizontal Dimension The length of the sampling line is X, the diagonal tiles for propagating the height of the double counter extend to at most and additional $\log X$, and the tiles for summation of all double counters can be implemented to extend to at most $\log XY$. The total length is thus at most $X + 2\log X + \log Y$.

Vertical Dimension The total height of the frame is at most $Y \log X$ for the line approximation, plus the space used to compute the ratio of the total sample length divided by the number of red tiles. The exact amount of space for division depends on the implementation, but a straightforward implementation of division using using tiles for simulating the execution of a Turing machine can divide an x bit input using space within an $x \times 3x^3$ box. Growing the larger dimension in the vertical direction and noting that the sum is at most $\log^3 XY$ yields a total length of $Y \log X + (\log XY)^3$.

5.2 Approximate Squares

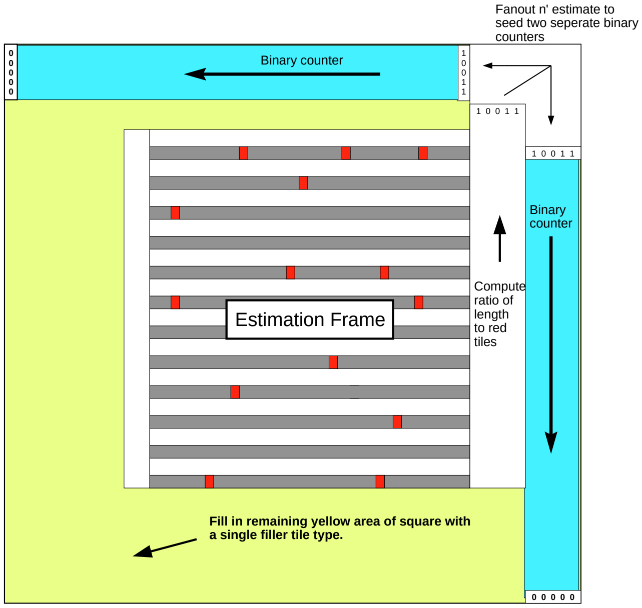
Fanout n' estimate to seed two seperate binary counters
Binary counter
0 0 0 0 0
1 0 0 1 1
1 0 0 1 1
1 0 0 1 1
Binary counter
Compute ratio of length to red tiles
Estimation Frame
Fill in remaining yellow area of square with a single filler tile type.
0 0 0 0 0

Fig. 4. This is a high level schemata for how the estimation frame can be used to build $n \times n$ squares.

Theorem 2 is the main theorem of this paper.

Theorem 2. *For any given $\epsilon, \delta \leq 1$ and $\frac{n}{\log n} \geq c\frac{log^3(1/\frac{\delta}{4})(1+\epsilon)^2}{\log \frac{1}{1-\frac{\delta}{4}}\epsilon^2}$, c a constant, there exists a tile system that with probability at least $1-\delta$ will assemble an $m \times m$ square with $(1-\epsilon)n \leq m \leq (1+\epsilon)n$.*

Proof. Consider a sufficiently large n. Let $n' = n - 2\log n$ and consider the approximation frame tile system to estimate n'. That is, set tile concentrations such that the probability q of placing a red tile for each of the sampling lines in the frame is $\frac{1}{n'}$. Let X denote the horizontal length of the sampling line growing east of the seed tile, and let Y denote the length of the line growing north of the seed tile (not counting the size $\log X$ buffer between each C tile).

First, we want to ensure that X and Y are short enough so that the entire frame will fit within the boundary of the target square. These bounds are $n_x = n - 4\log n$ for X and $n_y = \frac{n}{\log n} - 8\log^2 n - 1$ for Y. We also want to ensure that the total length of all sampling lines is sufficiently long to guarantee a good approximation. The following probability assignments ensure both constraints are met with high probability. Let the probability of placing the B and D tiles be $p_x = 1 - \frac{\delta}{4}^{\frac{1}{n_x}}$ and $p_y = 1 - \frac{\delta}{4}^{\frac{1}{n_y}}$, respectively.

First we show that $X \le n_x$ and $Y \le n_y$ with probability at least $1 - \frac{\delta}{2}$. Since X has a geometric distribution, we have that

$$\begin{aligned} P[X > n_x] &= (1-p)^{n_x}, \qquad (8)\\ &= (1-(1-(\frac{\delta}{4})^{\frac{1}{n_x}}))^{n_x}, \qquad (9)\\ &= \frac{\delta}{4} \quad \text{and,} \qquad (10)\\ P[Y > n_y] &= \frac{\delta}{4}. \qquad (11)\end{aligned}$$

Having shown that the frame will fit within the dimensions of the target square, we now show that the total length of all sampling lines XY will be sufficiently large under the assumption of a large n'.

Since X has the geometric distribution, we have that

$$\begin{aligned} P[X > n_x \frac{\log(1-\frac{\delta}{4})}{\log(\frac{\delta}{4})}] &= (1-(1-(\frac{\delta}{4})^{\frac{1}{n_x}}))^{n_x \frac{\log(1-\frac{\delta}{4})}{\log(\frac{\delta}{4})}} \qquad (12)\\ &= 1 - \frac{\delta}{4} \quad \text{and,} \qquad (13)\\ P[Y > n_y \frac{\log(1-\frac{\delta}{4})}{\log(\frac{\delta}{4})}] &= 1 - \frac{\delta}{4}. \qquad (14)\end{aligned}$$

Now consider the total length of the sample lines XY. With a similar analysis as is done for Theorem 1 we know that an (ϵ, δ)-approximations of n' can be achieved if the total length of the sample lines is

$$XY \ge \frac{3n'(1+\epsilon)^2 \log \frac{1}{\delta/4}}{\epsilon^2}. \tag{15}$$

From equations 13 and 14, we have that with probability greater than $1 - \frac{\delta}{2}$

$$\begin{aligned} XY &\ge n_x n_y (\frac{\log(1-\frac{\delta}{4})}{\log \frac{\delta}{4}})^2 \qquad (16)\\ &= \Omega(\frac{n^2}{\log n})(\frac{\log(1-\frac{\delta}{4})}{\log \frac{\delta}{4}})^2. \qquad (17)\end{aligned}$$

Combining equation 17 with inequality 15 shows that an (ϵ, δ)-estimate can be achieved in the case for a constant c where

$$\frac{n}{\log n} \ge c \frac{\log^3(1/\frac{\delta}{4})(1+\epsilon)^2}{\log \frac{1}{1-\frac{\delta}{4}} \epsilon^2}. \tag{18}$$

Given an (ϵ, δ)-approximation of n', it is straightforward to add a group of tiles that fanout the n' estimate into two identical copies which seed a binary counter. Each binary counter will begin counting down (decrementing rather than incrementing) until 0 is reached. The two counters then form the two axis of an $n' + 2\log n'$ square. Given this, it is straight forward to fill in the rest of the square at temperature 2 with a constant number of tiles. Thus, the approximation accuracy for the estimate n' yields a corresponding approximation accuracy for the dimension of the square assembled and thus proves Theorem 2. □

6 Future Work

This work is a preliminary theoretical look into the feasibility of precisely controlling assembled structures by manipulating tile concentrations. There are many directions for continued research.

One direction is to consider the probabilistic assembly of approximate scalings of general shapes. That is, given a tile system that assembles a given shape, modify the system so that a factor n magnification of the input shape is assembled. Along this line, it is should be possible to achieve (ϵ, δ)-approximate scalings with no increase in tile complexity for a large class of assembled shapes.

Another direction for future work is the design of probabilistic self-assembly systems for the assembly of exact shapes. That is, how can a tile system be designed so that a target shape is assembled exactly (no ϵ error factor) with high probability? It would be interesting to know if an alternate technique, or an improved version of our technique, could achieve this for $n \times n$ squares. An alternate approach would be to consider 3 dimension assemblies, such as $n \times n \times n$ cubes. In such a case, our approximation frame would be able to achieve a much higher asymptotic accuracy in n. With a tighter analysis, it is plausible that this could yield the exact assembly of cubes with high probability.

Yet another research direction involves the assembly of general shapes. In [13] a technique for the assembly of general scaled shapes is presented. To work, this technique first requires the assembly of a binary string of tiles encoding a description of the target shape. An interesting direction would be to combine this technique with the approximation frame from this paper to provide the input string assembled with $O(1)$ tile complexity. As the input string for the general shape system must be exact, a key step would be to ensure that the approximation frame is large enough to provide enough binary digits that contain no error (with high probability).

Finally, as our work here is theoretical, an important next step is simulation and lab experimentation to test and validate our results. Such experiments will likely provide key insights regarding how our model and technique can be improved.

References

1. L. Adleman, Q. Cheng, A. Goel, and M. Huang. Running time and program size for self-assembled squares. In *Proceedings of the 33nd Annual ACM Symposium on Theory of Computing*, pages 740–748, 2001.
2. L. Adleman, Q. Cheng, A. Goel, M. Huang, D. Kempe, P. Espanes, and P. Rothemund. Combinatorial optimization problems in self-assembly. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 23–32, 2002.
3. G. Aggarwal, Q. Cheng, M. H. Goldwasser, M.-Y. Kao, P. M. de Espanes, and R. T. Schweller. Complexities for generalized models of self-assembly. *SIAM Journal on Computing*, 34:1493–1515, 2005.
4. G. Aggarwal, M. H. Goldwasser, M.-Y. Kao, and R. T. Schweller. Complexities for generalized models of self-assembly. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 880–889, 2004.
5. F. Becker, E. Remila, and I. Rapaport. Self-assemblying classes of shapes with a minimum number of tiles, and in optimal time. In *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science*, 2006.
6. E. Demaine, M. Demaine, S. Fekete, M. Ishaque, E. Rafalin, R. Schweller, and D. Souvaine. Staged self-assembly: Nanomanufacture of arbitrary shapes with O(1) glues. In *Proceedings of the 13th International Meeting on DNA Computing*, 2007.
7. T.-J. Fu and N. C. Seeman. DNA double-crossover molecules. *Biochemistry*, 32:3211–3220, 1993.
8. M.-Y. Kao and R. Schweller. Reducing tile complexity for self-assembly through temperature programming. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithms*, pages 571–580, 2006.
9. T. H. LaBean, H. Yan, J. Kopatsch, F. Liu, E. Winfree, H. J. Reif, and N. C. Seeman. The construction, analysis, ligation and self-assembly of DNA triple crossover complexes. *J. Am. Chem. Soc.*, 122:1848–1860, 2000.
10. M. G. Lagoudakis and T. H. Labean. 2D DNA self-assembly for satisfiability. In *Proceedings of the 5th DIMACS Workshop on DNA Based Computers*, pages 459–468, June 26 1999.
11. J. Reif. Local parallel biomolecular computation. In *Proceedings of the 3rd Annual DIMACS Workshop on DNA Based Computers*, June 23-26 1997.
12. P. Rothemund and E. Winfree. The program-size complexity of self-assembled squares. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 459–468, 2000.
13. D. Soloveichik and E. Winfree. Complexity of self-assembled shapes. In *Tenth International Meeting on DNA Computing*, pages 344–354, 2005.
14. H. Wang. Proving theorems by pattern recognition. *Bell System Technical Journal*, 40:1–42, 1961.
15. E. Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology, Pasadena, 1998.

16. E. Winfree, F. Liu, L. Wenzler, and N. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, August 1998.
17. E. Winfree, X. Yang, and N. C. Seeman. Universal computation via self-assembly of DNA: Some theory and experiments. In *Proceedings of the 2nd International Meeting on DNA Based Computers*, pages 191–213, June 10-12 1996.