Time Optimal Self-assembly for 2D and 3D Shapes: The Case of Squares and Cubes

Florent Becker[1], Éric Rémila[1], and Nicolas Schabanel[2]

[1] Université de Lyon – LIP, UMR 5668 ENS Lyon CNRS UCBL
{florent.becker,eric.remila}@ens-lyon.fr
[2] CNRS, Universidad de Chile — CMM

Abstract. Self-assembling tile systems are a model for assembling DNA-based nano artefacts. In the currently known constructions, most of the effort is put on garanteeing the size of the output object, whereas the geometrical efficiency of the assembling of the shape itself is left aside. We propose in this paper a framework to obtain provably time efficient self-assembling tile systems. Our approach consists in studying how the flow of information has to circulate within the desired shape to guarantee an optimal time construction. We show how this study can yield an adequate ordering of the tiling process from which one can deduced a provably time efficient tile systems for that shape. We apply our framework to squares and cubes for which we obtain time optimal self-assembling tile systems.

Keywords: Self-assembling, Tilings, Time Optimal Construction, 2D and 3D Discrete Geometry.

1 Introduction

Self-assembly is the process by which small entities combine themselves into a bigger shape by local interactions in such a way that the resulting aggregates have interesting global properties. Examples of self-assembly are found in crystal growth, coral reefs, microtubules, and so on. In [7], Winfree proposed a framework for using DNA molecules in order to manufacture nano-artefacts by self-assembly. These artefacts can even be made to encode the result of a computation, leading to a good model for DNA computing [8].

This framework consists in a refinement of *Wang tiles* where the colors of the sides are seen as different kinds of *glues*. These glues have different *strengths* which represent the strength of the affinity between the DNA sequences forming the sides of the tiles. Two equal glues will stick together, with a strength equal to the strength of the glues. The self-assembly of the tiles is controlled by the *temperature*, an integer τ such that: a tile stays attached only if the sum of the strengths of the bonds with the other tiles along its sides is at least τ, otherwise it is torn off from the rest of the crystal by thermal agitation. This model can and has indeed been implemented practically [5] with DNA.

A. Goel, F.C. Simmel, and P. Sosík (Eds.): DNA 14, LNCS 5347, pp. 144–155, 2009.

Theoretical research so far on self-assembly has focused on minimising the numbers of tiles necessary to assemble a given shape. In this paper, we will focus as well on the time needed to assemble a shape. In this model, one can optimally assemble a shape with $O(K/\log K)$ tiles, where K is the Kolmogorov complexity of the shape [1,6]. This direction yields tile sets in which most of the complexity is related to the decoding of a compact representation of the shape, rather than assembling the shape efficiently. Furthermore, as far as we know, the time needed to assemble a given shape has not been studied in detail so far, and asymptotic results have been deemed sufficient. In this paper, we show how a finer understanding of the inherent parallelism of the model can lead to *time-optimal* self-assembly.

Time-constraint-based tile set design. Our approach relies on the study of the time-relationship between the tiles. More precisely, it consists in studying the flow of information within the desired shape, and deducing from it a specific order in which the tiles have to be placed, and from which we finally infer a time-optimal set of tiles. Deducing the tile set from its specifications allows to obtain directly a tile set whose behavior has already been proved formally, which is often easier than obtaining the tile set first and then proving that it behaves correctly. Interestingly enough, this method yields to a new kind of self-assembly process in which the construction is not driven by some master signals[1] but rather by a set of interdependent signals where no one takes over the other nor may progress without the others. This implies that other kinds of relationship between construction signals may have to be considered in future constructions.

On time optimality and tile set design. In order to set up a proper definition of time-optimal tile set for a shape, computer science taught us that the considered tile sets need to allow the construction of an infinite range of sizes for the object. Indeed, the optimal construction time of a *fixed size* object cannot be defined since its construction time can always be made arbitrarily small for any given tile set simply by increasing the concentrations of the tiles. Obtaining a tile set that can produce the desired shape at arbitrary size, is also a desirable design strategy: it's exactly like writing a program P that asks for n and computes $n!$ instead of rewriting a new program P_n each time a new n is requested. Decoupling the problem of decoding of the size and the problem of geometric assembling of the shape allows to focus on each problem separately and to treat each of them with more efficiency. Moreover, as we will see, our tile set can easily be extended by adding computations within the tiles that decodes the size and stops the construction at the right moment.

Our contribution. In this paper, we focus on squares and cubes and answer to the question: what is the optimal time to self-assembly these shapes? We start with the squares and design from the constraints imposed by time-optimality an order in which the tiles have to be placed, from which we deduce the tile set.

[1] For instance, in [2] the construction is driven by the master diagonal signal.

Our tile set allows to construct any square $n \times n$ in optimal parallel time $2n-2$ instead of $3n-5$ for the previously known constructions [2]. Then we extend the result to 3 dimensional self-assembly. In three dimensions, the description of the tile set could become cumbersome, and its verification even more so. We show how, by designing first the order in which the tiles should be added, and then by checking a few conditions of regularity and local determinacy on the order, we get at once the tile set and a proof of its correctness and its time optimality. One can also readily check properties such as the absence of *bubbles* in the 3D self-assembly process, which can be crucial in practice.

2 Self Assembling Tile Systems

Definition 1 (Tiles). *A* tile *is a unit square with one glue on each side. Formally, it is a quadruple of glues* $\langle \alpha_N, \alpha_S, \alpha_E, \alpha_W \rangle$ *chosen from a finite* set of glues Σ.[2] *Each glue* $\alpha \in \Sigma$ *has a* strength $g(\alpha)$*, a non-negative integer. Two tiles may be placed next to eachother only if their common sides hold the same glue* α*; in that case, we say that there is* bond *of strength* $g(\alpha)$ *between these tiles.*

Definition 2 (Tile system). *Formally, a* seeded tile system *is a quintuple* $\mathcal{T} = \langle T, t_0, \tau, \Sigma, g \rangle$*, where:*

- Σ *is a finite set of glues whose strengths are given by* g*.*
- $T \subseteq \Sigma^4$ *is a finite set of tiles.*
- $t_0 \in T$ *is a particular tile known as the* seed*.*
- τ *is a positive integer called the* temperature*.*

The "shapes" produced from a tile system are the one obtained by aggregation from the seed tile, according to the rule stating that a tile may stick to the current aggregate only if the sum of its bonds to the rest of the aggregate is at least the temperature. Formally,

Definition 3 (Productions). *A* configuration *is a partial mapping* $A : \mathbb{Z}^2 \to \mathcal{T}$*, such that all neighboring tiles hold the same glue on their common sides. The domain of* A *is called the* shape realized by A*. We say that a configuration* B extends *a configuration* A *at site* (i,j) *if: the domain of* B *is the domain of* A *plus the site* (i,j)*;* B *is identical to* A *on* A*'s domain; and the tile placed in* B *at site* (i,j) *sticks to* A*'s tiles, i.e., the sum of the strengths of the bonds around the sides of the tile* (i,j) *in* B *with the rest of the configuration is at least* τ*. We write* $A \to_{\mathcal{T}} B$ *and denote by* $\to^*_{\mathcal{T}}$ *the transitive closure.*

The productions $\mathcal{P}$ *of a tile system* $\mathcal{T}$ *are the configurations derived from the* seed configuration*,* I_0*, which consists in the single tile* t_0 *placed at* $(0,0)$*:*

[2] One can force the orientation of the tiles simply by coding the correct orientation in the glues, so we assume here w.l.o.g. that the tiles are *oriented* (no rotation of the tiles are allowed). This simplifies considerably the design of a tile system, without affecting its generality.

$\mathcal{P} = \{A : \Gamma_0 \rightarrow_{\mathcal{T}}^{*} A\}$. *We say that a production is* final *if no other production may be derived from it. The order induced by* $\rightarrow_{\mathcal{T}}^{*}$ *from the seed configuration is called the* dynamics *of* $\mathcal{T}$.

A desirable property of a tile system is that the tiles of a production have to be assembled in a fixed known (partial) order. This property turns out to be very useful in the proof of correctness of tile systems [5]. To the classic *RC condition* [5] which states that all the tiles on a given column (resp. row) have to be placed after a fixed first tile has been placed in this column (resp. row) (we say that this tile *opens* the column, resp. row), we prefer the following weaker condition which allows more geometrical freedom (for instance, assembling concave shapes) while preserving the necessary guarantee of determinism for the design and analysis purposes.

Definition 4 (The order condition). *Given an assembling sequence* $\gamma = \langle \Gamma_0 \rightarrow_{\mathcal{T}} A_1 \rightarrow_{\mathcal{T}} A_2 \rightarrow_{\mathcal{T}} \cdots \rightarrow_{\mathcal{T}} A_t = P \rangle$ *of a production* P *and two* neighboring *sites* (i, j) *and* (i', j') *in* P*, we say that* $(i, j) \prec_{P,\gamma} (i', j')$ *if the tile over* (i, j) *is placed before the tile over* (i', j') *in the construction sequence. We say that the tile system* $\mathcal{T}$ *satisfies the* order condition *(is* ordered*, for short) if for all production* P*, the relationship* $\prec_{P,\gamma}$ *between the sites of* P *is independent of the assembling sequence* γ*; in that case, the relationship, written* $\prec_P$*, defines a partial order (see for instance [3]) called the* local order *of the sites of* P.

Clearly, for any ordered tile system, the assembling sequence of a given production P is "locally deterministic": the tiles over two neighboring sites are always placed in the same order. The only uncertainty in an ordered tile system relies in the choice of the tile to place at each site; this choice will condition which production will be obtained at the end; but, every given production P is always assembled in the same order locally. Some easy facts on ordered tile systems are:

- if a production Q extends a production P, the local order $\prec_Q$ coincides with $\prec_P$ on P's sites, and moreover, since no site can be tiled before its predecessors, P is an ideal[3] $\prec_Q$ in the order theory terminology.
- since more than τ predecessors would induce uncertainty on the precedence relationship of a site with its neighbors, every site in a production P has at most τ predecessors according to $\prec_P$ and the seed t_0 is the only tile without predecessors;
- every RC tile system [5] is ordered.

3 Time Optimality and Skeleton

3.1 Time Optimal Tile Systems

Timed dynamics. The assembling of a shape is classically modeled by a Markov chain in which each tile appears at each unoccupied site at a rate proportional to its concentration [1]. We do not study here the probability that a given shape

[3] An *ideal* I of a partial order $\prec$ is a set such that for all $u \in I$, $v \prec u \Rightarrow v \in I$ (see [3]).

is produced, which depends on the concentrations of the tiles: we rather focus on minimizing the expected construction time of each production, conditioned to the event that this production is obtained. It is known from [1] that the expected construction time of a given production P is then proportional to the *length $\ell(P)$ of the longest chain of predecessors* in $\prec_P$, where the multiplicative constant depends only on the concentrations of the tiles. Since once we condition the expectation to construct one specific production, the concentrations play the same role as GHz in computers (doubling them will reduce the construction time by 2), the meaningful parameter to measure time here is $\ell(P)$. We thus aim at *minimizing the length of the longest chain of dependencies in $\prec_P$* to obtain an efficient construction time. Therefore, we say that the construction of a final production P is *time optimal* when the length $\ell(P)$ of the longest chain in $\prec_P$ is as small as possible.

Note that, for ordered tile systems, $\ell(P)$ is also exactly the time to construct P in the *parallel dynamics*, where at every time step, we place all the tiles that can be attached to the current aggregate.

Since any production is obtained by assembling tiles one after the other starting at the seed tile, we obtain the following bound on the length of the longest chain of dependencies.

Fact 1 (Trivial lower bound). *Given a 4-connected shape $S \subset \mathbb{Z}^2$, for all tile system $\mathcal{T}$ and all production P realizing S, we have: $\ell(P) \geqslant ||S||_1$, where $||S||_1$ denotes the maximum ℓ_1-distance of a site of S to the seed tile placed at the origin.*

Definition 5 (Real time assembling). *Given a family of shapes $\mathcal{S}$, we say that a tile system realizes the family $\mathcal{S}$* in real time*, if the shapes realized by its final productions are exactly $\mathcal{S}$, and for all final production P realizing a shape $S \in \mathcal{S}$, we have: $\ell(P) = ||S||_1$.*

By Fact 1, any real time tile system is by definition time optimal. The following definition extends the notion of time to every site of a production in an ordered tile system.

Definition 6 (Rank). *We define the* rank $\rho_P(u)$ *of a site u in a production P, as the length of the longest chain leading to u in $\prec_P$. We say that a site u of P is* on time *if $\rho_P(u) = ||u||_1$. Clearly, a tile system realizes a production P in real time, if the site of highest rank in P is on time. Note that in the parallel dynamics assembling a production P, each site u is tiled at time $\rho_P(u)$ exactly.*

3.2 Skeleton of a Production: Lower Bounding the Rank

Existing self-assembling constructions are driven by main signals, the rest of the production been completed by "passive" tiles filling the gaps between these signals. In fact, this has to be the case for all ordered self-assembling constructions and we formalize this fact with the following notion:

Definition 7 (Skeleton). *Let $\mathcal{T}$ be an ordered tile system. The* x-skeleton *of a production P (resp., y-*skeleton*) is the union of the seed site and of the sites of P which have a unique predecessor in $\prec_P$, placed on their right or left (resp., above or below). The* skeleton *of P is the union of its x- and y-skeletons.*

The x-skeleton (resp. y-) of a production P is the set of sites that *open* the x-dimension (resp. y-) during the assembling of P, i.e., which are the first in each column (resp., row) to be tiled. Indeed, in an ordered tile system, each column of a production is progressively tiled by agregation from the corresponding x-skeleton site on this column. Note that even if the tiles aggregated on a column may carry as well meaningful signals, they have all to be placed by agregation from the x-skeleton tile on this column. Naturally, the same holds respectively for rows and the y-skeleton. It follows that given an ordered tile system, one can lower bound the rank $\rho_P(u)$ of each site u in a production P by the maximum of the following two quantities:

$$\sigma_X(u) = \rho_P(a) + ||u - a||_1 \quad \text{and} \quad \sigma_Y(u) = \rho_P(b) + ||u - b||_1, \tag{1}$$

where a and b are the closest x- and y-skeleton sites on the column and on the row of u in P, respectively (i.e., where a and b are the sites which open the row and the column of u, respectively, in the assembling of P).

These two lower bounds show how the "flow of information" circulates within the shape from the skeleton during the assembling. We will see next how one can use these lower bounds to understand *where* the skeleton of a shape has to be to match the real time constraints. We illustrate this approach by obtaining two real time tile systems realizing, respectively, the $n \times n$ squares S_n and the $n \times n \times n$ cubes C_n, rooted at the origin, in optimal time.

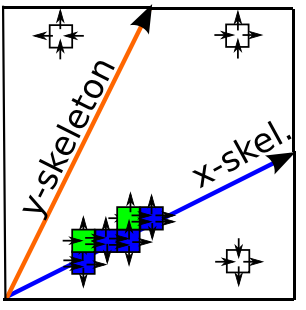
4 Assembling Squares in Real Time

4.1 A Real Time Local Order for Squares

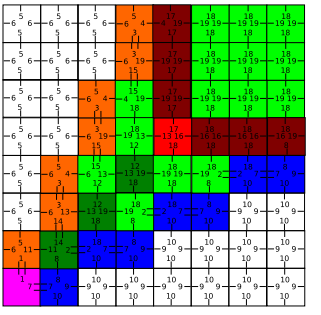
We first show how to assemble in real time the family of squares with lower left corner at the origin, $S_n = \{0, \ldots, n-1\}^2$ for $n \geqslant 2$. Our approach consists in designing first a local order for the sites from the "flow of information" induced by the real time constraints within the shape; and then deduce from it an ordered tile systems matching this order.

Since $||S_{n+1}||_1 = 2n$, our goal is to obtain a local order for the square S_{n+1} such that the highest rank of the sites is (at most) $2n$. Consider an arbitrary ordered tile system $\mathcal{T}$ that realizes the square S_{n+1}. Since S_{n+1} is convex and $\mathcal{T}$ is ordered, each column (resp., row) of S_{n+1} contains exactly one site in the x-skeleton (resp., y-); otherwise, if there were two x-skeleton sites on the same column, the sites between them could be assembled from both below and above, the local order would depend on the assembling sequence and $\mathcal{T}$ would not be ordered. Let us thus denote $a_i = (i, y_i)$ and $b_j = (j, x_j)$ the sequences of sites in the x- and y-skeletons respectively. If $y_n > n/2$ (for instance in [2,4], $y_n = n-1$), then by the lower bound (1), the rank of lower right corner $(n, 0)$ would be at least $\rho(a_n) + ||(n, 0) - (n, a_n)||_1 \geqslant ||a_n||_1 + y_n = n + 2y_n > 2n$. It follows that the x-skeleton (resp. y-) of a real time ordered tile system for S_{n+1} cannot get above the site $(n, \lfloor n/2 \rfloor)$ (resp. to the left of the site $(\lfloor n/2 \rfloor, n)$).

We thus consider the following x- and y-skeletons: $a_i = (i, \lfloor i/2 \rfloor)$ and $b_j = (\lfloor j/2 \rfloor, j)$ as displayed in Figure 1(a). The lower bound (1) tells us that the



(a) The local order induced by the x- and y-skeletons within the square. In each of the three regions delimited by the x- and y-skeletons, the local order $\prec$ follows the directions of the arrows. For example, in the central region, the predecessors of (i, j) are $(i-1, j)$ and $(i, j-1)$ and its successors are $(i+1, j)$ and $(i, j+1)$. Each site $(i, \lfloor \frac{i}{2} \rfloor)$ (resp. $(\lfloor \frac{j}{2} \rfloor, j)$) of x-skeleton (resp. y-) has a unique predecessor, $(i-1, \lfloor \frac{i}{2} \rfloor)$ (resp., $(\lfloor \frac{j}{2} \rfloor, j-1)$).



(b) The tile system for even squares. The numbers on the sides of each tile are the IDs of the glues, and the number of ticks accross the boundary of the tiles stands for the strength of the glue. The temperature is 2. Strength 2 glues lie on the lines $j = 2i$ and $j = i/2$. The green tiles and the red tile on the diagonal represent the GO and STOP tiles which propagate (on time) up to the blue and orange signals carrying the x- and y-skeletons respectively either to allow them to continue their progression for the next two steps (GO) or to stop them (STOP).

Fig. 1. The skeleton and the tile systems assembling squares in real time

rank of each site $u = (i, j)$ in any local order based on this skeleton is at least: $\sigma(u) = \max(||a_i||_1 + ||u - a_i||_1, ||b_j||_1 + ||u - b_j||_1)$. More precisely,

- $\sigma(u) = i + j = ||u||_1$ if u lies between the x- and y-skeletons, i.e., if $j \geqslant \lfloor i/2 \rfloor$ and $i \geqslant \lfloor j/2 \rfloor$;
- $\sigma(u) = i + 2\lfloor i/2 \rfloor - j > ||u||_1$ if u lies below the x-skeleton, i.e., if $j < \lfloor i/2 \rfloor$;
- symmetrically, $\sigma(u) = j + 2\lfloor j/2 \rfloor - i > ||u||_1$ if u lies above the y-skeleton, i.e., if $i < \lfloor j/2 \rfloor$.

Clearly, $\sigma(u) \leqslant 2n$, for all u. Moreover, one can easily verify that σ is the rank of the following local order where the predecessors of each site $u = (i, j)$ are defined as (see Figure 1(a)):

- its east neighbor $(i-1, j)$, if $u = a_i$ belongs to the x-skeleton;
- its south neighbor $(i, j-1)$, if $u = b_j$ belongs to the y-skeleton;
- its east and north neighbors $(i-1, j)$ and $(i, j+1)$, if u lies below the x-skeleton;
- its east and south neighbors $(i-1, j)$ and $(i, j-1)$, if u lies between the x- and y-skeletons;
- its west and south neighbors $(i+1, j)$ and $(i, j-1)$, if u lies above the y-skeleton.

We are now left with implementing this real time order with an ordered tile system.

4.2 A Real Time Tile System for Squares

The main issue in implementing the real time local order above into a tile system is the *synchronization* between the two *distant* signals carrying the x- and y-skeletons. It turns out that the sites lying between the x- and y-skeletons *are all on time* (their ranks equal their ℓ_1-norms). We may then use this zone to transport just-in-time the synchronization signals. We proceed as shown in Figure 1(b). The progress of the signals carrying the x- and y-skeletons is controlled by the tile placed on the diagonal: if the GO tile is placed on the diagonal, then the GO signal propagates along the row and column and allows just-in-time the x- and y-skeletons to jump by one upward and to the right respectively and to progress for two new steps; if the STOP tile is placed on the diagonal, then the STOP signal propagates along the row and column and the progression of the x- and y-skeleton stops just-in-time forcing the production to be completed into a square. In order to obtain even square as well, we have two kinds of STOP tiles: EVEN and ODD; while EVEN stops where the x- and y-skeleton stoped, ODD allows one single extra step to extend the length of the sides by one.

The complete description of the tiles and their proof of correctness cannot be given in details due to space constraints but can be read easily in Figure 1(b). We thus conclude:

Theorem 2. *There exists an ordered self-assembling system with 24 tiles (with rotations allowed) whose final productions at temperature* 2 *realize exactly the set of all squares S_n in optimal time.*

Note that the tile system proposed in Figure 1(b) is not minimal and one can significantly reduce the number of tiles by shifting the x- and y-skeletons while preserving time optimality. The classic counter-based technique in the literature [4] which allows to control precisely the size of the assembly can also be used with this construction. It consists in adding $O(\log n/\log\log n)$ tiles to force the production of an $n \times n$ square. Instead of choosing non-deterministically the position of the STOP tile on the diagonal, one can use the on time region between the x- and y-skeletons to simulate a counter (or any other computation) and have the (ODD or EVEN) STOP tile to be put exactly at $(\lfloor n/2 \rfloor, \lfloor n/2 \rfloor)$. Moreover, using the same technique, and by launching simultaneously four real time constructions with a counter in the directions SE, NE, NW, and SW, one can also obtain a real time assembly of a square rooted on any of its sites.

5 Assembling Cubes in Real Time

5.1 The Skeleton

We will now show how our framework allows to obtain readably a tri-dimensional tile system which assemble in optimal time the family of cubes rooted at the origin, $C_n = \{0, \ldots, n-1\}^3$ for $n \geqslant 2$. We extend canonically the definitions in Sections 2 and 3 from 2D to 3D.

As observed in Section 3.2, the assembling of a convex shape by an ordered tile systems is necessarily driven by three skeletons consisting in the sequence

of sites opening each dimension. Taking inspiration from the 2D tile system designed above, we consider the following x-, y- and z-skeletons heading from the origin to the centers of the three opposite faces of the cube: $a_i = (i, \lfloor i/2 \rfloor, \lfloor i/2 \rfloor)$, $b_j = (\lfloor j/2 \rfloor, j, \lfloor j/2 \rfloor)$ and $c_k = (\lfloor k/2 \rfloor, \lfloor k/2 \rfloor, k)$.

As before, we define the *rank function σ induced by the skeleton* as follows. For each site $u = (i, j, k)$, let $\sigma_X(u) = ||a_i||_1 + ||u - a_i||_1$, $\sigma_Y(u) = ||b_j||_1 + ||u - b_j||_1$, and $\sigma_Z(u) = ||c_k||_1 + ||u - c_k||_1$. The rank of u induced by the skeleton is then: $\sigma(u) = \max(\sigma_X(u), \sigma_Y(u), \sigma_Z(u))$. Since tiles are assembled from the skeleton by definition of the skeleton, $\sigma(u)$ is clearly a lower bound on the earliest time a tile may be placed at u in the parallel dynamics.

Given two neighboring sites u and v in $\mathbb{N}^3$, we say that u is a *predecessor* of v (and v is a *successor* of u) if $\sigma(u) < \sigma(v)$. We define the *order induced by the skeleton* $\prec$ as the transitive closure of the predecessor relation: $u \prec v$ if there exists a finite sequence $(u_0 = u, u_1, \ldots, u_q = v)$ such that u_{i-1} is a predecessor of u_i for all $i = 1, \ldots, q$. Our goal is to prove that there exists a tile system whose local order is exactly $\prec$. We need first to understand the structure of this order, and thus first, the local variations of σ.

5.2 On the Rank Function Induced by the Skeleton

We denote by (e_x, e_y, e_z) the canonical basis of $\mathbb{N}^3$. A careful case study yields the following key lemma describing the variations of σ along the z-axis.

Lemma 1 (Variations of σ along the z-axis). *For all site $u = (i, j, k)$ in $\mathbb{N}^3$, if $\max(i, j) \leqslant 2k + 1$, then $\sigma(u) < \sigma(u + e_z)$; and if $\max(i, j) \geqslant 2k$, $\sigma(u) < \sigma(u - e_z)$ (provided that $k \neq 0$).*

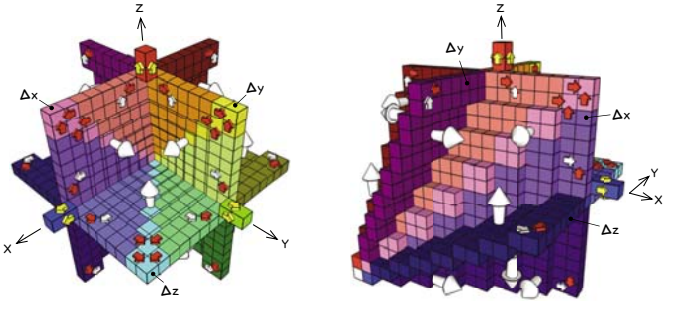


Fig. 2. The local variations of σ and the corresponding local order in $\mathbb{N}^3$ from two points of views: from (n, n, n) on the left, and from $(n, 0, n)$ on the right. Only the sites with at most 2 predecessors are represented. Arrows points in σ's growth direction along the axes. The white arrows stand for simple predecessors; the red arrow for double predecessors; and the yellow arrows for triple predecessors. One can clearly see the partition in seven regions filled with sites with 3 simple predecessors, separated by discrete planes made of sites with two predecessors including at least one double predecessor, intersecting each other on the x-, y- and z-skeletons (in pure blue, green and red respectively) made of sites with one single triple predecessor.

We thus define the following "pyramidal" discrete surfaces: $\Delta_X = \{(\lfloor \frac{\max(j,k)}{2} \rfloor, j, k) : j, k \geqslant 0\}$, $\Delta_Y = \{(i, \lfloor \frac{\max(i,k)}{2} \rfloor, k) : i, k \geqslant 0\}$ and $\Delta_Z = \{(i, j, \lfloor \frac{\max(i,j)}{2} \rfloor) : i, j \geqslant 0\}$. One can reformulate Lemma 1 as follows: σ is an increasing function of z above Δ_Z, and decreasing function of z below Δ_Z.

Figure 2 sums up this result graphically. The discrete surfaces Δ_X, Δ_Y and Δ_Z are respectively painted in blue and pink colors, green and orange colors, and blue and green colors (the meaning of the colors will be explained in the following sections). The growth directions of σ along each axis are represented by the large white arrows within each of the seven regions partitioned by these three surfaces and by small arrows within each surface.

Two important facts follows. First, σ defines a "proper" local order: σ never takes the same value on two neighboring sites and thus, for every pair of neighboring sites, one is always the predecessor one of the other. Second, the highest ranked site in each cube C_{n+1} is $3n = ||C_{n+1}||_1$, which is attained at the corner (n, n, n). It follows that the order induced by the skeleton matches the real time constraints as expected.

5.3 Classification of the Sites According to Their Relative Positions with Their Predecessors

Thanks to Lemma 1 and Figure 2, we observe the following properties of the order $\prec$ induced by σ:

Fact 3

1. *each site has at most three predecessors among its neighbors;*
2. *the origin is the unique site with no predecessor;*
3. *a site u has a unique predecessor if and only if u belongs to the skeleton; this predecessor is $u - e_z$ if u lies in the z-skeleton.*
4. *every site $u = (i, j, k)$ with exactly two predecessors has two opposite neighbors which are its successors; and these opposite successors are located in $u \pm e_z$ if and only if u belongs to Δ_Z.*

Multiplicity. We define the *multiplicity* of the predecessors of the sites as follows (the missing cases are obtained by symmetry). The multiplicity will be used later on to define the strength of the bond between the tiles attached on these sites.

- The multiplicity of the predecessors of any site with three predecesors is 1;
- If $u \neq 0$ belongs to the skeleton, the multiplicity of its unique predecessor is 3;
- If $u = (i, j, k)$ belongs to Δ_Z: if $\lfloor \frac{i}{2} \rfloor \geqslant \lfloor \frac{j}{2} \rfloor$ (u belongs to the bluish part of Δ_Z), the multiplicity of its predecessor along the y-axis is 2; if $\lfloor \frac{j}{2} \rfloor \geqslant \lfloor \frac{i}{2} \rfloor$ (u belongs to the greenish part of Δ_Z), the multiplicity of its predecessor along the x-axis is 2; if both cases apply (u belongs to the diagonal sites in cyan of Δ_Z in Figure 2), both predecessors of u get a multiplicity of 2, otherwise the other predecessor gets a multiplicity 1.

In Figure 2, the multiplicities of the predecessors with their successors are represented by white, red or yellow arrows whether the multiplicity is 1, 2, or

3 respectively. The following fact will ensure that the tile system we will derive from the order in Section 5.5, assembles the productions in the prescribed order at temperature 3.

Fact 4. *For all site $u \neq 0$, the sum of the multiplicities of its predecessors is at least* 3 *and the sum of the multiplicities of any strict subset of its predecessors is at most* 2*.*

5.4 Deducing the Successors from the Predecessors

A key property needed to obtain the tile system is that the information transmitted by the glues on the faces a site u shares with its predecessors, is enough, independently of the position of u, to completely determine the correct glues of the free faces of u. In terms of order, this property consists for now in checking that one can deduce the correct types of the successors of u from the types of its predecessors only, and a small number of criteria that can be transmitted from site to site through the glues. Consider a site $u = (i, j, k)$.

- If u has a triple predecessor, w.l.o.g., $u - e_x$, then $u = a_i$ belongs to the x-skeleton, u has four double successors $u \pm e_y$ and $u \pm e_z$, and: if i is even, then $u + e_x = a_{i+1}$ is a triple successor, and a simple successor otherwise.
- If u has two double predecessors, w.l.o.g., $u - e_x$ and $u - e_y$ (u belongs to the cyan diagonal part of Δ_Z), then $u \pm e_z$ are both simple successors, and: if i (resp. j) is even, $u + e_x$ (resp. $u + e_y$) is a double successor, and a simple successor otherwise.
- If u has two predecessors, a double and a simple, w.l.o.g., its double and simple predecessors are respectively $u - e_x$ and $u - e_y$ and u belongs to the lighter blue part of Δ_Z; then $u \pm e_z$ are simple successors and $u + e_x$ and $u + e_y$ are respectively double and simple successors as well.
- If u has three simple predecessors, then either u has three simple successors or it has one triple successor, w.l.o.g. a_{i+1}, and two simple successors; the second alternative arising only if $u = a_i + e_y + e_z$, i.e., if u is a successor of $a_i + e_y$ which is itself a successor of a_i.

Each of these cases corresponds to a different color in Figure 2.

5.5 A Real Time Tile Systems for Cubes

It follows from the study above that one can correctly guess the types of the successors of any site from the only knowledge of: 1) the multiplicity and relative position of its predecessors, 2) the parity of the corresponding coordinates, and 3) possibly the immediate proximity to a part of the skeleton. We define thus the glues as a combination of three labels as follows:

- each face holds a label chosen among $++$, $+$, $-$, $--$, $---$ according to the multiplicity of the predecessor/successor relation with the corresponding neighboring site and according to the orientation of this relation with respect to the corresponding axis.

- each pair of opposite faces holds a 0 or a 1 to propagate the parity of each coordinate.
- each even site a_{2i} of the x-skeleton propagates to $a_{2i} + e_y$ which in turn relays to $a_{2i} + e_y + e_z$ the fact that the later has to allow a triple successor on its x^+-face to attach the next element a_{2i+1} of the x-skeleton (the same holds for the y- and z-skeletons).

According to the previous subsections, we now have a set of tiles that agregate themselves from the seed tile according to the desired local order $\prec$ induced by σ. We are now left with the problem of the synchronization of the stop of the three distant signals carrying the x-, y- and z-skeletons. As for the squares in Section 4.2, we can use for that purpose the central region between the three surfaces Δ_X, Δ_Y and Δ_Z. Indeed an easy calculation demonstrates that this region consists of the union of all the $q \times q \times q$ cubes $\{q, \ldots, 2q\}^3$ and that its sites are all on time (recall definition 6) and can thus be used to propagate from the main diagonal a GO, STOP-EVEN, STOP-ODD signal to the three part of the skeleton simultaneously and without delay as we did before in the 2D case. We can thus conclude:

Theorem 5. *There exists an ordered self-assembling system with a finite number of tiles whose final productions at temperature* 3 *realize exactly the set of all cubes C_n in optimal time.*

References

1. Adleman, L.M., Cheng, Q., Goel, A., Huang, M.-D.A.: Running time and program size for self-assembled squares. In: STOC 2001: Proc. of the 33rd ACM symposium on Theory of computing, pp. 740–748 (2001)
2. Becker, F., Rapaport, I., Rémila, E.: Self-assemblying classes of shapes with a minimum number of tiles, and in optimal time. In: LNCS Proc. of Found. of Software Tech. and Theo. Comp. Sci., pp. 45–56 (2006)
3. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, Cambridge (2002)
4. Rothemund, P.W.K., Winfree, E.: The program-size complexity of self-assembled squares (extended abstract). In: STOC 2000: Proceedings of the 32nd symp. on Theory of computing, pp. 459–468 (2000)
5. Rothemund, P.W.K.: Theory and Experiments in Algorithmic Self-Assembly. PhD thesis, University of Southern California (2001)
6. Soloveichik, D., Winfree, E.: Complexity of self-assembled shapes. In: Ferretti, C., Mauri, G., Zandron, C. (eds.) DNA 2004. LNCS, vol. 3384, pp. 344–354. Springer, Heidelberg (2005)
7. Winfree, E.: Algorithmic Self-Assembly of DNA. PhD thesis, Caltech (1998)
8. Winfree, E.: Simulations of computing by self-assembly. In: Proceedings of the Fourth DIMACS Meeting on DNA Based Computing, pp. 213–242 (1998)