Natural Computing manuscript No.
(will be inserted by the editor)

The Effect of Malformed Tiles on Tile Assemblies within the kinetic Tile Assembly Model

Ya Meng · Navin Kashyap

Received: date / Accepted: date

Abstract Many different constructions of proofreading tile sets have been proposed in the literature to reduce the effect of deviations from ideal behaviour of the dynamics of the molecular tile self-assembly process. In this paper, we consider the effect on the tile assembly process of a different kind of non-ideality, namely, imperfections in the tiles themselves. We assume a scenario in which some small proportion of the tiles in a tile set are "malformed". We study, through simulations, the effect of such malformed tiles on the self-assembly process within the kinetic Tile Assembly Model (kTAM). Our simulation results show that some tile set constructions show greater error-resilience in the presence of malformed tiles than others. For example, the 2- and 3-way overlay compact proofreading tile sets of Reif et al. (2005) are able to handle malformed tiles quite well. On the other hand, the snaked proofreading tile set of Chen and Goel (2005) fails to form even moderately-sized tile assemblies when malformed tiles are present. We show how the Chen-Goel construction may be modified to yield new snaked proofreading tile sets that are resilient not only to errors intrinsic to the assembly process, but also to errors caused by malformed tiles.

Keywords kinetic Tile Assembly Model · malformed tiles · snaked proofreading tile sets

1 Introduction

Molecular self-assembly is a process of bottom-up fabrication of complex structures from simple parts. The tile assembly models introduced by Erik Winfree (Winfree 1998a,b) are widely used mathematical models of this process. The models suitably extend Wang's tiling model (Wang 1961) by taking into account some of the thermodynamic aspects of molecular self-assembly. The building blocks in this model are square tiles with labels and "glues" on the edges. In the *abstract Tile Assembly Model (aTAM)*, tiles attach to each other along edges with matching labels, provided the strength of the attachment (determined by the glues) exceeds a certain threshold. Tiles attaching to each other according to these rules of attachment form large assemblies. This process can be used to carry out computation, by encoding data and computational rules in the tile edge labels and glues.

Y. Meng · N. Kashyap
Department of Mathematics and Statistics, Queen's University, Kingston, ON, K7L 3N6, Canada
E-mail: nkashyap@mast.queensu.ca

The *kinetic Tile Assembly Model (kTAM)* augments aTAM with a stochastic model of self-assembly dynamics, yielding a more realistic model of molecular self-assembly. The dynamics of kTAM allow tiles to attach to each other in violation of the attachment rules of aTAM. This can result in assembly errors, *i.e.*, departures from ideal growth of the assembly. One means of controlling such assembly errors is through *proofreading tile sets*, constructions of which have been suggested by Winfree and Bekbolatov (2004), Chen and Goel (2005), Reif et al. (2005), and Soloveichik and Winfree (2006). Each of these constructions has its strengths and weaknesses, as discussed by Soloveichik and Winfree (2006).

In this paper, we consider a completely different type of error, one arising not due to imperfections in the process of assembly, but rather due to imperfections in the process of tile set fabrication. We assume a scenario in which, at the time of fabricating tiles out of DNA molecules, some small proportion of the tiles is imperfectly created. This could happen, for example, due to the denaturing of molecules caused by chemical or environmental effects, or simply due to human error in the fabrication process. We use the term *malformed tiles* to denote these imperfectly created tiles. We consider a model of malformed tile creation in which some of the tile edges receive labels different from the designed (correct) labels. We then ask the following question: how do tile sets containing a small proportion of malformed tiles deviate from their designed assembly behaviour under kTAM? To the best of our knowledge, the behaviour of tile assemblies in the presence of malformed tiles has not been previously studied in the literature.

We present extensive simulation results comparing error rates in assemblies with and without malformed tiles. In our simulations, carried out within the Winfree group's tile assembly simulator, `xgrow`[1], we considered a range of different tile sets, including several different constructions of proofreading tile sets. As expected, assembly-error rates in tile sets with malformed tiles exceeded those in tile sets without such tiles. Also predictably, the extent of resilience to malformed tile errors varies from one tile set to another. Most notably, the snaked proofreading tile set of Chen and Goel (2005) performed very poorly in simulations. In the presence of malformed tiles, this tile set showed a tendency to fatally stall, failing to even form moderately sized tile aggregates[2]. While looking for an explanation for this phenomenon, we discovered two new snaked proofreading tile set constructions that counter the effect of malformed tiles quite well. These new constructions are significant because they retain the "snaked" order of growth of the original Chen-Goel construction, which also makes them resistant to growth and facet errors intrinsic to the self-assembly process.

The rest of the paper is organized as follows. In Section 2, we provide brief descriptions of aTAM and kTAM, and of the various proofreading tile set constructions we consider in this paper. Section 3 describes our model for malformed tiles, Section 4 contains our simulation results, and Section 5 discusses malformed snaked proofreading tile sets and our new constructions of snaked proofreading tile sets. We make some concluding remarks in Section 6.

2 Background

We provide here a brief description of Winfree's tile assembly models sufficient for our purposes; for more details, the reader may refer to any of several detailed explanations of

[1] Available online at `http://www.dna.caltech.edu/Xgrow`

[2] In this paper, we use the terms "aggregate" and "aggregation" to mean an assembly of tiles.

the model that exist in the literature (Chen and Goel 2005; Reif et al. 2005; Rothemund et al. 2004; Winfree 1998b).

2.1 The abstract and kinetic Tile Assembly Models

The basic building block of aTAM is a square *tile*, which has four edges, and which belongs to one of finitely many *tile types*. The collection of tile types allowed in the model is called the *tile set*. A tile is not allowed to be rotated, and hence, the four sides of a tile have fixed directions — in clockwise order, north, east, south, west — assigned to them. Two edges, not necessarily belonging to the same tile, are called *opposite edges* if they are either a north-south pair or an east-west pair.

Associated with each tile is a *tile label* taken from some finite alphabet. The edges of tiles may be given *edge labels*, and each edge label has a non-negative integer called the *glue strength* associated with it. The tile label, edge labels, and glue strengths for a tile are completely determined by the tile type to which it belongs. Two opposite edges from distinct tiles are called *matching edges* when they both have the same edge label. Matching edges can attach to each other to form *bonds*, the strength of the bond being equal to the glue strength of either edge in the matching pair forming the bond. Bonds of strength 0, 1, and at least 2, are called *null*, *weak*, and *strong* bonds, respectively. Assembly occurs by the iterative addition of tiles to an existing aggregate, initialized by a special *seed tile*. In aTAM, a tile can be legally added to a position at the border of an existing aggregate if the total strength of the bonds formed by the matching edges in that addition is greater than or equal to a system parameter called *temperature*, which is always a positive integer. The arrangement of tile labels obtained from an assembly of tiles is called a *pattern*.

Figure 1 shows an example of the *basic Sierpinski tile set*. This tile set consists of four kinds of *rule tiles*, two kinds of *boundary tiles*, and one seed tile. If the system temperature is set to 2, which means that each addition of a new tile to an existing aggregate requires at least two weak bonds or one strong bond, then upon initialization by the seed tile, the Sierpinski tile set forms the Sierpinski *pattern* shown in Figure 2. This assembly consists of a "reverse-L" shaped frame consisting of the seed tile at the corner and boundary tiles along the arms of the reverse L, to which the rule tiles attach forming the internal part of the assembly. The assembly grows in a bottom-right to top-left fashion, determining an orientation on the tiles in the tile set. The orientation of a rule tile is from its *input sides* to its *output sides*, as depicted by the dashed arrows in Figure 1. The input sides of each rule tile are the right (east) and bottom (south) edges, while the output sides are the top (north) and left (west) edges. By convention, a boundary tile has only one input side and one output side; the edges labelled b in boundary tiles, as depicted in the figure, are not considered to be input/output sides. The basic Binary Counter tile set (Figure 3) is another simple tile set which has the same number of rule and boundary tiles as the Sierpinski tile set.

The kTAM is an augmentation of aTAM that includes a *forward rate* for tiles to associate to the growing assembly, and a *reverse rate* for tiles to detach themselves from the assembly. The forward rate of attachment of a tile depends only on the overall concentration of tiles in the ambient solution, and is given by $r_f = ke^{-G_{mc}}$, where G_{mc} is the negative logarithm of tile concentration, and k is a constant that sets the time scale. The rate at which a tile dissociates from the assembly depends on how strongly the tile is attached to the assembly. The dissociation rate is given by $r_{r,b} = ke^{-bG_{se}}$, where k is the same constant as above, b is the total strength of the bonds between the tile and the assembly, and G_{se} denotes the free energy of breaking a single bond. Thus, for example, a tile that attaches itself to an existing

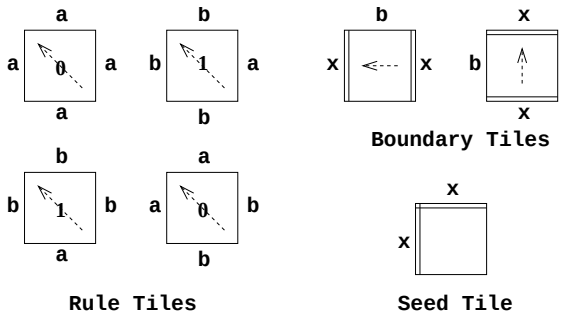


Fig. 1 The basic Sierpinski tile set. The edge labels with glue strength 1 are denoted by a or b. Double lines denote edges with glue strength 2, all of which have edge label x. Tile labels (0/1) have been placed at the centre of each rule tile. The orientation of a tile from its input sides to its output sides is depicted by a dashed arrow.

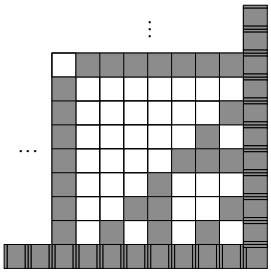


Fig. 2 An initial fragment of the Sierpinski pattern formed by the basic Sierpinski tile set. The seed tile, boundary tiles, and rule tiles labelled '1' have been shaded grey, while rule tiles labelled '0' have been left unshaded.

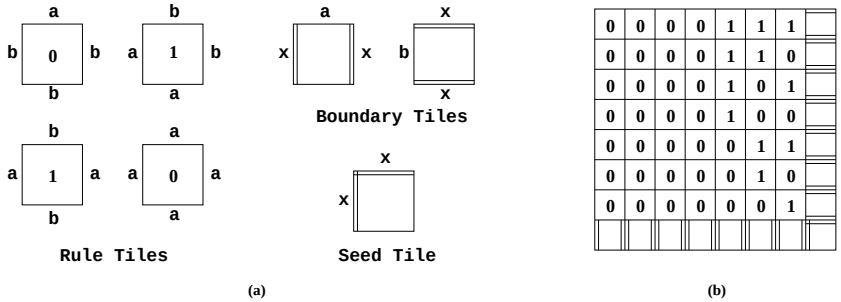


Fig. 3 (a) The basic Binary Counter tile set. The edge labels with strength 1 are denoted by a or b. Double lines denote edges with glue strength 2, all of which have edge label x. Tile labels (0/1) have been placed at the centre of each rule tile. (b) An initial fragment of the Binary Counter pattern formed by the tile set in (a).

assembly with a total bond strength of 2 is more likely to dissociate from the assembly than a tile attaching to the assembly with bond strength 3. This observation plays a key role in our analysis of the phenomenon of stalling observed in certain tile assemblies, which we discuss in Section 5.1.

If we set $\frac{G_{mc}}{G_{se}} = b - \epsilon$, for small $0 < \epsilon < 1$, and consider a tile attaching to a position with a total bond strength of b, we will have $\frac{r_f}{r_{r,b}} = e^{\epsilon G_{se}} > 1$, so that tile addition is more likely to happen than not. On the other hand, if we consider a tile attaching to a position

with a total bond strength of $b-1$, we will have $\frac{k_f}{k_{r,b}} = e^{-(1-\epsilon)G_{se}} \ll 1$, and this addition is not likely to occur. Based on this, one may reasonably infer that the ratio $\frac{G_{mc}}{G_{se}}$ plays the same role in kTAM as the system temperature in aTAM. Thus, an attachment that is legal in aTAM is also favored to happen in kTAM.

However, it can happen within kTAM (but not within aTAM) that a tile attaches to the perimeter of the aggregate with insufficient bond strength, but before it falls off, another tile attaches next to it as a legal addition, resulting in a stable assembly overall. The initial insufficient attachment may be due to mismatched edges (*growth error*), or may be due to the attachment of an isolated tile at a facet of the aggregate (*facet error*[3]). Such errors remain "frozen" within the assembly, and may cause further errors, resulting in a final assembly that differs from the designed (correct) final assembly.

Recall that the arrangement of tile labels in an aggregate forms a pattern. A pattern corresponding to an error-free aggregate is a *correct pattern*. The arrangement of tile labels in an aggregate that contains errors may still form the correct (designed) pattern, but it is more likely that such an aggregate results in a *wrong pattern*, *i.e.*, an arrangment of tile labels that is different from the designed pattern.

2.2 Proofreading tile sets

Proofreading tile sets are a means of controlling errors in the tile assembly process. Loosely speaking, the idea behind such schemes is to start with a basic tile set (such as the Sierpinski tile set in Figure 1), and modify it by introducing redundant information into the edge labels of tiles, which can then act as an error-control mechanism. The first example of such a tile set construction was due to Winfree and Bekbolatov (2004), who replaced each tile of the basic tile set with k^2 proofreading tiles arranged in a $k \times k$ block. The edge labels internal to each $k \times k$ block are all unique to that block. Figure 4(b) shows an example of a 2×2 proofreading construction. However, the overall error rate for this scheme did not scale well with k, because such proofreading tile sets, while being able to control growth errors, were largely ineffective against facet errors. Chen and Goel (2005) introduced the $k \times k$ *snaked proofreading tile set* that modified the Winfree-Bekbolatov construction by changing the glue strengths on some of the internal edges of the constituent tiles within each $k \times k$ block. By appropriately introducing strength-0 and strength-2 edges into the constituent tiles, they were able to control the manner in which the constituent tiles assembled to form a $k \times k$ block. This allowed them to successfully deal with facet errors (as well as growth errors), and they were able to show that error rates in their assemblies decreased exponentially with k. Figure 4(c) shows the construction of a 2×2 snaked proofreading block.

Both the constructions mentioned above reduce error rates at the cost of scaling up the size of the final assembly by a factor of k^2. Reif et al. (2005) showed that certain basic tile sets (such as the Sierpinski and Binary Counter tile sets) could be transformed into *compact* proofreading tile sets that lowered error rates without scaling up the size of the final assembly. They gave two specific constructions — a "2-way overlay" construction and a "3-way overlay" construction — in which each edge label of a proofreading tile encodes two or three bits of partially redundant information. For the Sierpinski and Binary Counter patterns, the 2-way and 3-way overlay constructions resulted in tile sets with 8 rule tiles and 16 rule tiles, respectively. The constructions of Reif et al. do not appear to be easily generalizable to an "n-way overlay" construction.

[3] This was called "nucleation error" by Chen and Goel (2005).

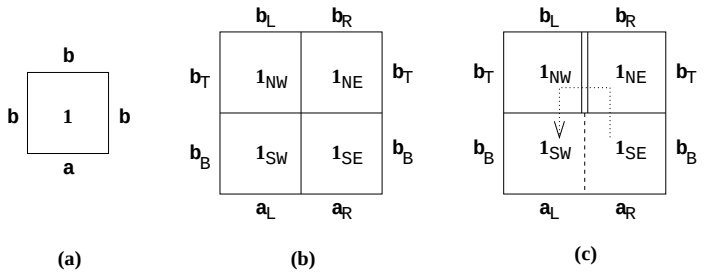


Fig. 4 2×2 proofreading tile sets: (a) Original tile (from basic Sierpinski tile set); (b) Winfree-Bekbolatov construction; (c) Chen-Goel snaked construction, with order of growth of the 2×2 block depicted by a dotted line. Dashed lines, single lines, and double lines depict edges with glue strength 0, 1, and 2, respectively.

Soloveichik and Winfree gave a different construction, which they attribute to Paul Rothemund and Matthew Cook (see Soloveichik and Winfree 2006, Appendix B), of a general n-way overlay compact proofreading tile set, in which each edge label of a proofreading tile encodes n bits of information. Meng (2009) has shown that for the Sierpinski pattern, the n-way overlay construction results in a tile set with $n^2 + 3n + 4$ rule tiles. It should again be stressed that, while the number of tiles in any of these compact proofreading tile sets is larger than that in the original Sierpinski or Binary Counter tile set, the size of the final assembly required to produce an $N \times N$ Sierpinski or Binary Counter pattern is exactly the same as that for the original tile set.

3 Malformed Tile Sets

As stated in the introduction, the object of this paper is to consider a problem different from the one tackled by proofreading tile set constructions. The scenario that motivates us is one in which errors affect the process of tile set fabrication, resulting in abnormalities or imperfections in some small proportion of the tiles. Thus, in our set-up, malformed tiles are simply deformations of normal tiles. In fact, since the only sort of tile abnormalities that influence the assembly process are imperfections in tile edges, we will assume that abnormal tiles are created by modifying the edges of normal tiles.

So, suppose that we are given a *normal tile set* $\mathfrak{T}$ consisting of tile types that we consider to be *normal tile types*. A *malformed tile type* is obtained by modifying some (or all) of the edge labels of some normal tile type. The edges in the malformed tile type whose labels differ from the normal edge labels will be called *malformed edges*. A *malformed tile set* derived from $\mathfrak{T}$ is a collection of tile types consisting of the (normal) tile types in $\mathfrak{T}$ along with some malformed tile types derived from the tile types in $\mathfrak{T}$.

We now describe certain simplifying assumptions we make in our model for malformed tile sets. Our first assumption is based on an understanding that edge imperfections do not transform one normal edge label into another normal edge label.

Assumption 1 *Malformed edges always receive labels that are distinct from all the edge labels of normal tile types.*

We do not specify at this point whether or not distinct malformed edges can receive the same edge label. We will consider two extreme cases in our simulations — one in which no pair of distinct malformed edges receives the same edge label, and another in which all

malformed edges receive identical edge labels. In the former case, any malformed tile in the interior of a tile assembly must have a mismatched edge with at least one of its neighbours.

The next assumption is consistent with the fact that malformed tile types are obtained from normal tile types only through edge modifications.

Assumption 2 *A malformed tile type has the same tile label as the normal tile type from which it is formed.*

Note that this assumption allows for the possibility that a tile assembly based on a malformed tile set may contain mismatched edges and still produce the same (correct) pattern as an assembly obtained from the normal tile set. We will see the effect of Assumption 2 being manifested in our simulation results in the next section.

Assumption 3 *A malformed tile type contains exactly one malformed edge.*

The rationale behind Assumption 3 is that, among malformed tiles, those with more than one malformed edge have a negligible impact on the assembly process, as they are unlikely to form stable attachments with an existing aggregate.

The definitions and assumptions we have made up to this point remain valid for an arbitrary normal tile set. However, to make our subsequent discussions easier, we will from now on restrict ourselves only to the normal tile sets covered by our simulations, namely, the basic Sierpinski tile set (Figure 1), the basic Binary Counter tile set, and various proofreading tile sets derived from these basic tile sets. There are three kinds of tile types in these tile sets — a unique seed tile, boundary tiles, and rule tiles.

We will assume that only rule tiles can be modified to form malformed tiles. This is because malformed edges on the seed tile or on boundary tiles either do not influence the tile assembly process at all or cause the process to come to a standstill. To illustrate this more clearly, we take the example of the Sierpinski tile set in Figure 1. Note that imperfections in the strength-0 edges of the seed or boundary tiles would have no influence on the tile assembly process. On the other hand, a malformed strength-2 edge in a seed tile or boundary tile would make the formation of a stable assembly highly unlikely. Boundary tiles in this tile set also have strength-1 edges, imperfections in which would have an influence on the assembly process. However, the effect of such a malformed edge can be equivalently modelled by modifying the matching strength-1 edge in the rule tile that must attach to it. We are thus justified in making the following assumption.

Assumption 4 *Malformed tiles are obtained by modifying rule tiles only.*

Rule tiles have two input edges (bottom and right) and two output edges (top and left). To differentiate between the effects of malformed input edges and malformed output edges, we categorize malformed tile sets into three classes:

(1) *Input-side malformed (IM) tile sets*, in which the malformed tile types consist of all tile types obtainable by modifying one input edge of a normal rule tile.
(2) *Output-side malformed (OM) tile sets*, in which the malformed tile types consist of all tile types obtainable by modifying one output edge of a normal rule tile.
(3) *Input/output-side malformed (IOM) tile sets*, in which the malformed tile types consist of all possible IM tile types and OM tile types.

Figure 5 contains a depiction of these malformed tile types. In our simulations, we deal separately with each of the above classes of malformed tile sets.

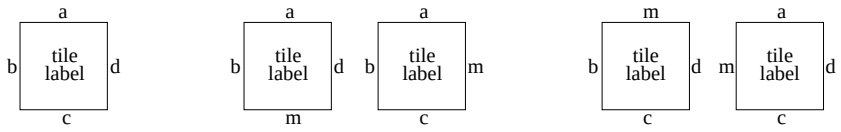


Fig. 5 The tile on the left is a normal rule tile, the two tiles in the middle represent IM tile types, and the two tiles on the right represent OM tile types. Malformed edges are denoted by the label m.

4 Simulation Results

The simulations results reported in this section provide a broad comparison of the performance of various kinds of tile sets in the presence of malformed tiles. The tile sets we have simulated produce either the Sierpinski pattern or the Binary Counter pattern. For the Sierpinski pattern, our simulation results cover the following tile sets: the basic Sierpinski tile set (Figure 1), Winfree and Bekbolatov's 2×2 proofreading tile set (Winfree and Bekbolatov 2004), the 2- and 3-way overlay compact proofreading tile sets of Reif et al. (2005), and the general n-way overlay construction attributed to Rothemund and Cook (Soloveichik and Winfree 2006, Appendix B). While we also ran simulations for the snaked proofreading tile set of Chen and Goel (2005), we found this tile set to perform very poorly in the presence of malformed tiles, as we explain in more detail in the next section. The tile sets covered in our simulations for the Binary Counter pattern are: the basic Binary Counter tile set (Figure 3), the 2×2 proofreading tile set, and the 2- and 3-way overlay compact proofreading tile sets of Reif et al. We did not run simulations for malformed tile sets derived from the Rothemund-Cook n-way overlay construction for the Binary Counter. The reason for this is that, for $n = 1, 2, 3$, the error rate obtained with the n-way overlay tile set, even without malformed tiles, is larger than that for the basic Binary Counter tile set. While the error rates do decline as n increases, the number of tiles in these tile sets increases exponentially with n, so that for $n \geq 4$, the number of tiles is large enough that it becomes infeasible to simulate the corresponding malformed tile sets within `xgrow`.

All simulations were performed within the `xgrow` tile assembly simulator. The values of the parameters G_{mc} and G_{se} were set to 17 and 8.6, respectively, in keeping with the principle expounded by Winfree (1998b) that the ratio of G_{mc} to G_{se} should be kept slightly below 2. For each type of rule tile, the tile concentration value was set to [0.9]; for each malformed tile type, the tile concentration value was set to [0.05]; the seed tile concentration was [0.1], and the concentration of each boundary tile type was [0.3]. We ran 50 simulations for each tile set. Unless explicitly specified otherwise, our malformed tile sets have the property that malformed edges in distinct malformed tile types receive distinct edge labels (see the remarks following Assumption 1).

Table 1 A comparison of the number of errors in a 128×128 assembly, averaged over 50 runs.

	Basic tile set	IM tile set	OM tile set	IOM tile set
Sierpinski	7.36	8.14	9.78	11.74
Binary Counter	7.16	8.62	14.82	15.38

Table 1 above compares the performance of the basic Sierpinski and Binary Counter tile sets to that of the malformed tile sets derived from them. The performance measure here is the number of errors in a 128×128 tile assembly, averaged over 50 runs. As expected, malformed tile sets perform worse than the normal tile sets, Furthermore, tile sets containing

Table 2 A comparison of various malformed proofreading tile sets for the Sierpinski pattern. The first number in each table entry is the proportion, out of 50 tile aggregations, of aggregations that contain mismatched-edge errors. The second number, which is in parentheses, is the proportion of aggregations that fail to produce a correct 128×128 block of the Sierpinski pattern.

	Normal tile set	IM tile set	OM tile set	IOM tile set
Basic Sierpinski tile set	1 (1)	1 (1)	1 (1)	1 (1)
2×2 proofreading	0 (0)	0.56 (0.02)	0.76 (0.44)	0.94 (0.48)
Reif et al. 2-way overlay	0 (0)	0.3 (0.06)	0.42 (0.14)	0.58 (0.2)
Reif et al. 3-way overlay	0 (0)	0.18 (0)	0.24 (0.04)	0.30 (0.04)
Rothemund-Cook 1-way overlay	0 (0)	0.26 (0.06)	0.34 (0.1)	0.58 (0.16)
Rothemund-Cook 2-way overlay	0 (0)	0.36 (0)	0.38 (0.02)	0.52 (0)
Rothemund-Cook 3-way overlay	0 (0)	0.26 (0)	0.44 (0.02)	0.56 (0.02)

OM tile types perform worse than IM tile sets, a trend that is consistent across all simulation results reported in this section. This phenomenon has a straightforward explanation. Consider any open position in the growing asembly where a normal tile can stably attach itself (by forming two weak bonds). An OM tile derived from the same normal tile has an equal chance of attaching itself to that position, as the input edges of the malformed tile are the same as those of the normal tile. On the other hand, an IM tile has one malformed input edge, so its attachment to the open position must happen with insufficient bond strength. This makes the attachment of the IM tile less likely to occur in kTAM than the attachment of a normal tile. This explains why the proportion of malformed tiles attaching to an aggregation in an OM or IOM tile set is higher than that in an IM tile set. However, we are unable to satisfactorily explain why the performance gap between the IM and the OM/IOM tile sets is much larger for the Binary Counter pattern than for the Sierpinski pattern.

The influence of malformed tiles on various proofreading tile set constructions for the Sierpinski pattern is tabulated in Table 2. The entries in the table attempt to measure the ability of several different malformed tile sets to form a 128×128 block of the Sierpinski pattern. In the case of malformed tile sets derived from the 2×2 proofreading tile set, this requires the formation of a 256×256 aggregate of tiles; but in all other cases, only a 128×128 tile aggregate is required. Each entry of the table consists of a decimal number followed by another such number in parentheses. The first number gives the proportion, out of a total of 50 aggregations, of aggregations that contain mismatched-edge errors. The second number (the one in parentheses) represents the proportion, out of 50, of aggregations that result in a wrong 128×128 pattern. Recall that, by virtue of Assumption 2, it is possible for an assembly formed from malformed tile sets to produce the correct pattern even if the assembly contains mismatched edges. Thus, the second number in any of the entries in the Table 2 can never exceed the first number in that entry. In fact, in many of the entries, the second number is much lower than the first. This is especially true of the malformed tile sets derived from the Rothemund-Cook 2- and 3-way overlay tile sets.

Among the malformed tile sets considered in Table 2, those derived from the 2×2 proofreading tile set produce the least number of error-free aggregations. This is entirely reasonable, since these tile sets are required to produce larger assemblies than the other tile sets listed in the table.

The most number of error-free aggregations is produced by the 3-way overlay tile set of Reif et al. Interestingly, the malformed Rothemund-Cook 2-way and 3-way overlay tile sets are remarkably consistent in producing correct 128×128 blocks of the Sierpinksi pattern almost all the time. This observation induced us to study the performance of these tile sets over a larger assembly size of 256×256 tiles. We restricted our attention to the worst-

Table 3 A comparison of IOM tile sets derived from the Rothemund-Cook n-way overlay tile sets for the Sierpinski pattern. Target aggregate size is 256×256.

	$n = 1$	$n = 2$	$n = 3$	$n = 4$
Average number of errors in a 256×256 aggregate	5.28	4.02	3.06	3.12
Proportion of aggregates with errors	0.98	0.96	0.92	0.94
Proportion of aggregates forming wrong patterns	0.84	0.4	0.22	0.32

Table 4 A comparison of various malformed proofreading tile sets for the Binary Counter pattern. The first number in each table entry is the proportion, out of 50 tile aggregations, of aggregations that contain mismatched-edge errors. The second number, which is in parentheses, is the proportion of aggregations that fail to produce a correct 128×128 block of the Binary Counter pattern.

	Normal tile set	IM tile set	OM tile set	IOM tile set
2×2 proofreading	0 (0)	0.76 (0.02)	0.86 (0.34)	0.98 (0.36)
Reif 2-way overlay	0 (0)	0.26 (0.04)	0.34 (0.06)	0.5 (0.1)
Reif 3-way overlay	0 (0)	0.24 (0)	0.28 (0.04)	0.52 (0.04)

case IOM tile sets only. Fifty (50) simulation runs were performed on each of the IOM tile sets derived from the Rothemund-Cook n-way overlay construction, for $n = 1, 2, 3, 4$. The results of these simulations are presented in Table 3. These results show that despite the fact that almost all the 256×256 tile aggregates produced by these IOM tile sets have mismatched-edge errors in them, the proportion of aggregates that fail to produce the correct 256×256 Sierpinski pattern block is (at least for $n = 2, 3, 4$) much smaller.

It is interesting to note from the results presented in Table 3 that the malformed n-way overlay tile sets improve in performance as n increases from 1 to 3, but the improvement stops at $n = 3$. Two things happen in the normal n-way overlay tile sets as n increases:

(a) the amount of redundant information encoded in each edge label increases, and
(b) the total number of tile types increases — as mentioned in Section 2.2, the tile set has $n^2 + 3n + 4$ rule tiles.

These two factors seem to affect the performance of the derived malformed tile sets in opposite ways. Increased redundancy in the edge labels appears to mitigate, to some extent, the effect of malformed tiles. On the other hand, the number of tile types in a normal tile set seems to adversely affect the performance of malformed tile sets derived from it. Indeed, evidence can be found even in Table 2 for the fact that the number of tile types in a normal tile set plays a role in determining the performance of derived malformed tile sets. The 2-way overlay tile set of Reif et al., and the Rothemund-Cook 1-way overlay tile set have eight tiles each, and the malformed tile sets derived from them display remarkably similar performance.

Turning now to Binary Counter tile sets, we provide in Table 4 a performance comparison analogous to that in Table 2 for Sierpinski tile sets. As mentioned at the beginning of this section, we did not run simulations for malformed tile sets derived from the Rothemund-Cook n-way overlay construction for the Binary Counter. Thus, Table 4 only compares the malformed tile sets derived from the 2×2 proofreading tile set, and the 2-way and 3-way overlay compact proofreading tile sets of Reif et al. The target aggregate size is 256×256 for the malformed 2×2 proofreading tile sets, and 128×128 for the others. The simulation results again indicate that the proportion of assemblies producing the correct pattern is a much more useful indicator of malformed tile set performance than the proportion of assemblies having mismatched edges.

Our last collection of simulation results compares the performance of malformed tile sets in which malformed edges from distinct tile types receive distinct labels to those in which

Table 5 The number of mismatched-edge errors in a 128 $\times$ 128 tile assembly, averaged over 50 runs.

	Basic Sierpinski IOM tile set	Basic Binary Counter IOM tile set
Distinct malformed edges receiving distinct labels	11.74	15.38
Distinct malformed edges receiving same labels	9.64	10.40

Table 6 The proportions of aggregates containing errors, and (within parentheses) of aggregates producing wrong patterns, for IOM tile sets derived from the Rothemund-Cook n-way overlay tile sets for the Sierpinski pattern. The proportions are calculated over 50 aggregates, each of size 128 $\times$ 128.

	$n = 1$	$n = 2$	$n = 3$
Distinct-label IOM tile set	0.58 (0.16)	0.52 (0)	0.56 (0.02)
Same-label IOM tile set	0.70 (0.82)	0.60 (0.12)	0.50 (0.08)

all malformed edges receive the same label. We refer to the former variety of malformed tile sets as *distinct-label*, and the latter as *same-label*. While a same-label tile set is probably not a very realistic model for malformed tiles, it will serve as a control for our experiments on distinct-label malformed tile sets. Observe that we only need to focus on IOM tile sets here, as it is only these tile sets that can benefit (or not) from the potential of a malformed input edge in one tile matching with a malformed output edge of another. Note also that if malformed input and output edges receive distinct edge labels (as happens in a distinct-label tile set), then the glue strengths assigned to those edges play no role within kTAM. However, if malformed input and output edges are allowed to have the same edge labels, then it does matter what glue strength is assigned to those labels. If all pairs of matching malformed edges have glue strength 0, then the resulting effect within kTAM is the same as that of distinct-label tile sets. So, in our simulations, to be able to truly distinguish the effect within kTAM of same-label IOM tile sets from that of distinct-label IOM tile sets, we set the glue strength to be equal to 1 for all malformed edges in our same-label tile sets.

The results of these simulations are presented in Tables 5 and 6. As usual, the results are for 50 simulation runs for each tile set shown in the tables, and a target assembly size of 128 $\times$ 128 tiles. Table 5 indicates that the number of mismatched-edge errors in a tile assembly is smaller, on average, for same-label IOM tile sets than for distinct-label ones, as one would expect. On the other hand, it can be seen from Table 6 that same-label tile sets tend to produce more incorrect patterns than distinct-label tile sets. This is readily explained by the fact that in assemblies formed by distinct-label tile sets, a tile attaching itself to a position in the growing assembly is much more likely than not to be either the correct tile for that position, or a malformed tile derived from the correct tile. Tiles other than these have a much lower chance of being "frozen" within the assembly by the addition of further tiles. Thus, tiles attaching to positions in the growing assembly are highly likely to have the correct tile label for that position, which makes it quite probable that the final pattern produced by the assembly is correct.

5 Snaked Proofreading Tile Sets

At the beginning of the previous section, we mentioned the fact that malformed tile sets derived from the snaked proofreading tile sets of Chen and Goel (2005) performed very poorly in simulations. We investigate this phenomenon in detail in this section, and propose some solutions to the problem.

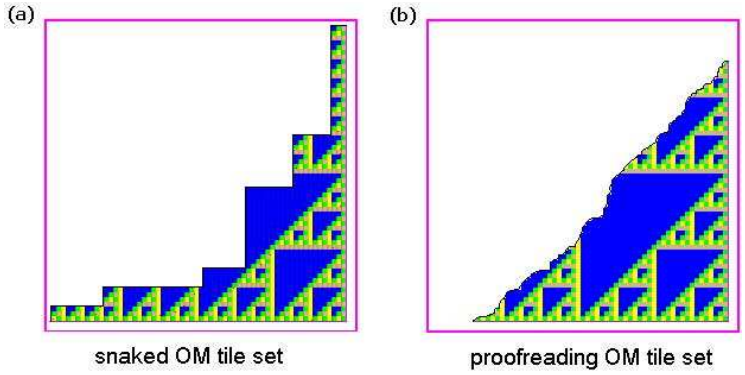


Fig. 6 `Xgrow` simulation screen shots of (a) a 4×4 snaked proofreading OM tile set, and (b) a 4×4 (non-snaked) proofreading OM tile set. In both cases, the target assembly size is 64×64.

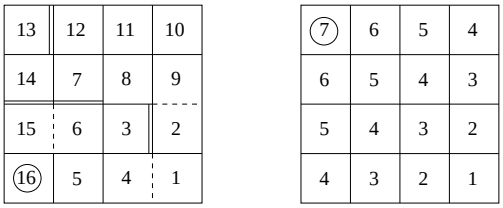
5.1 Stalling in Malformed Snaked Proofreading Tile Assemblies

The simulations we undertook focused on malformed tile sets obtained from the 4×4 snaked proofreading tile set generating the Sierpinski pattern. On the one hand, IM tile sets were able to form 256×256 aggregations, but these aggregations were riddled with mismatched-edge errors. On the other hand, OM and IOM tile sets failed to even produce aggregations of this size. In our simulations, OM tiles caused the tile assembly process to stall repeatedly and frequently.

It is natural to compare the snaked proofreading tile set with the Winfree-Bekbolatov (non-snaked) proofreading tile set (Winfree and Bekbolatov 2004), as the former is basically a modification of the latter. We ran simulations on OM tile sets derived from 4×4 proofreading tile sets of both kinds. Under identical simulation conditions, the Winfree-Bekbolatov 4×4 proofreading OM tile set was able to produce a complete 256×256 tile aggregation in a very short time, while the 4×4 snaked proofreading OM tile set was unable to form an aggregation of this size even after several hours of uninterrupted simulation runs.

We provide here an explanation for the stalling phenomenon described above. Figure 6 shows two screen shots of the growing assemblies of a 4×4 (non-snaked) proofreading OM tile set and a 4×4 snaked proofreading OM tile set. Note that the growth front of the proofreading OM tile set is smooth, whereas that of the snaked OM tile set is uneven and jagged. From careful observations of the manner in which these assemblies grow, we found that tile addition in a snaked proofreading assembly occurs in a block-by-block fashion, while in a proofreading assembly, the addition of new tiles is not strictly correlated to the formation of blocks. More precisely, in a snaked proofreading assembly, once a new 4×4 block starts forming, further tiles are added in a manner that favours the completion of that block over the initiation of the growth of other blocks (see Figure 6(a)). Furthermore, once a block is completed, that block tends to remain stable, with only a very small chance of tiles falling off from that block according to the dynamics of kTAM. On the other hand, in a (non-snaked) proofreading assembly, the growth front constantly renews itself through the process of tiles adding and falling off, and this is largely independent of the formation of complete 4×4 blocks.

We noticed that stalling in an OM snaked proofreading tile assembly almost always occurred in situations when a complete 4×4 block was formed, but there were OM tiles frozen within the output-side (top and left) edges of the complete block. The frozen OM



(a) (b)

Fig. 7 4×4 proofreading constructions: (a) the Chen-Goel snaked proofreading construction, (b) the Winfree-Bekbolatov (non-snaked) construction. Dashed edges, single edges, and double edges depict strength-0, strength-1, and strength-2 bonds, respectively. The numbering of the tiles shows the order in which tiles attach to form the 4×4 block. The circles identify the last tiles to attach in a block.

tiles hindered the formation of the next 4×4 block, and it took an exceedingly long time for them to fall off according to the dynamics of kTAM, causing the stall. The reason for this is that, as noted by Chen and Goel (2005), all the tiles on the output sides of a 4×4 snaked proofreading block (see Figure 7(a)) are held by bonds of total strength at least 3. So, whenever such a block is completely assembled, it is unlikely that any of the tiles within the block will fall off, and be replaced by other tiles. In particular, if any of the tiles on the output sides of a completed 4×4 block are OM tiles (with malformed edges facing outside), these are likely to remain frozen within the block. Note that this is not as much of a problem with the (non-snaked) proofreading tile assemblies, where at any time during the formation of a 4×4 block, there are always tiles within the block that are attached to the rest of the assembly with bonds of total strength at most 2; see Figure 7(b).

5.2 New Snaked Proofreading Constructions

To alleviate the problem of stalling, we propose two new snaked proofreading constructions, referred to as Construction A and Construction B, which are depicted in Figure 8(a) and Figure 8(b), respectively. Both constructions follow the same principles as the Chen-Goel construction, but differ from the latter construction in the positions of the null bonds and strength-2 bonds. Construction A changes some of the outer edges in the Chen-Goel 4×4 snaked proofreading block to null bonds, and some of the inner edges to strength-2 bonds. Construction B is the same as the Chen-Goel snaked proofreading construction within the bottom-right 3×3 sub-block. The novelty in the construction is that the tiles in the fourth row and column of a block have only strength-1 edges. The idea of this construction is to combine a 3×3 snaked proofreading construction with a row and a column of the non-snaked proofreading construction.

Note that in both the new constructions, the last tile to attach within a block (identifiable by the circles drawn in Figure 8) is held to the assembly by bonds of total strength 2. Thus, if the assembly were to stall after a complete 4×4 block is formed, the chances of the last tile falling off — and hence, allowing other tiles to fall off as well — are higher in these constructions than in the Chen-Goel construction.

As with any proofreading constructions that rely upon the formation of 4×4 blocks, Constructions A and B can certainly reduce growth errors. Construction A also preserves the "snaked" order of growth of the Chen-Goel construction, which has been shown to be effective in countering the influence of facet errors (Chen and Goel 2005). Construction B preserves the snaked order of growth of the Chen-Goel construction only within the first

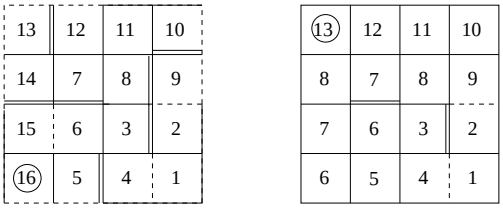


Fig. 8 4×4 Snaked proofreading constructions: (a) Construction A, (b) Construction B. Dashed edges, single edges, and double edges depict strength-0, strength-1, and strength-2 bonds, respectively. The numbering of the tiles shows the order in which tiles attach to form the 4×4 block. The circles identify the last tiles to attach in a block.

3×3 sub-block. As a result, it too can reduce facet errors, but perhaps not as much as the Chen-Goel construction and Construction A.

The advantage of Construction B over Construction A is that it uses a lot fewer null bonds and strength-2 bonds, particularly on the outer edges of a block. We have observed in our simulations that the self-assembly process finds it easier to assemble 4×4 blocks of Construction B than of Construction A.

We report here the results of our simulations comparing the two new snaked proofreading constructions with the Chen-Goel construction. All our simulations were run in `xgrow` under the growth conditions obtained by setting $G_{mc} = 17$ and $G_{se} = 8.6$.

The first experiment compares assemblies formed by the normal tile sets for the three snaked proofreading constructions. Each normal tile set was able to successfully complete a 256×256 block of tiles (*i.e.*, a 64×64 block of the Sierpinski pattern). Tabulated below is the number of correct aggregates (and thus correct patterns) formed in 50 simulation runs by the normal tile sets for the three constructions.

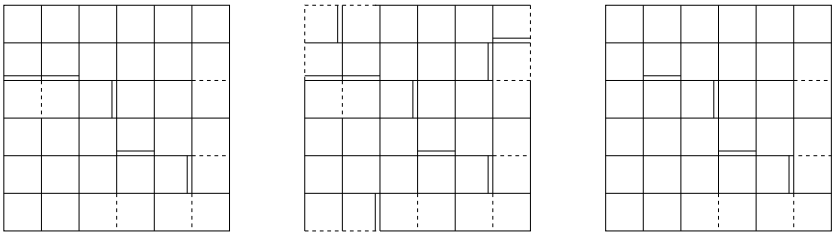
Chen-Goel	Construction A	Construction B
50	43	48

It is interesting to note that while the two new snaked proofreading constructions are not always able to form correct aggregates, Construction B is more consistent at forming a correct aggregate than Construction A.

We next compare the performances of the OM tile sets obtained from the three constructions. The concentration of each malformed tile is equal to $[.005]$, compared to a concentration of $[0.9]$ for each normal rule tile. Fifty simulation runs were performed on each OM tile set, with a target assembly size of 256×256 tiles. Simulations were run in batches of 25 parallel runs on a standard desktop computer, with each batch allowed to run uninterrupted for 30 hours. The entries in the table below record the number of runs, out of 50, that result in a complete 256×256 tile assembly, and (within parentheses) the number of runs that result in a complete and correct 64×64 block of the Sierpinski pattern.

Chen-Goel	Construction A	Construction B
0 (0)	30 (29)	36 (33)

It is clear from the simulation results that the two new snaked proofreading constructions significantly improve upon the performance of the Chen-Goel construction in the presence of OM tiles. In particular, they are able to avoid the stalling phenomenon observed in the Chen-Goel snaked proofreading OM tile assembly. Also, a large proportion of the complete Construction A and Construction B aggregates yield correct Sierpinski patterns.



(a) **(b)** **(c)**

Fig. 9 Snaked proofreading constructions of 6×6 blocks: (a) Chen-Goel construction, (b) Construction A, (c) Construction B. Dashed, single, and double lines represent strength-0, strength-1, and strength-2 bonds, respectively.

The new 4×4 snaked proofreading constructions presented above can be generalized to $2k \times 2k$ blocks, for any $k \geq 2$. In the descriptions that follow, $T_{i,j}$ denotes the tile in row i and and column j of a $2k \times 2k$ block. Here, rows are indexed $1, 2, \ldots, 2k$, starting with the bottom row, and the columns are indexed $1, 2, \ldots, 2k$, starting with the rightmost column.

The $2k \times 2k$ Construction A snaked proofreading block is described as follows:

1. The west sides of tiles $T_{1,2i-1}$ are null bonds, where $i = 1, 2, \ldots, k-1$.
2. The north sides of tiles $T_{2i,1}$ are null bonds, where $i = 1, 2, \ldots, k-1$.
3. Edges on the north sides of tiles $T_{2i,2i+1}$ have strength 2, where $i = 1, 2, \ldots, k-1$.
4. Edges on the west sides of tiles $T_{2i,2i-1}$ have strength 2, where $i = 1, 2, \ldots, k$.
5. The edges on the north sides of tiles $T_{2k-1,1}$ and $T_{2k-2,2k}$ have strength 2.
6. The edges on the west sides of tiles $T_{2k-1,1}$ and $T_{1,2k-2}$ have strength 2.
7. The west side of tile $T_{2k-2,2k-1}$ is a null bond.
8. The north sides of tiles $T_{2k,2k-1}$ and $T_{2k,2k}$ are null bonds.
9. The west sides of tiles $T_{2k-1,2k}$ and $T_{2k,2k}$ are null bonds.
10. The south sides of tiles $T_{1,2k-1}$ and $T_{1,2k}$ are null bonds.
11. The east sides of tiles $T_{2k-1,1}$ and $T_{2k,1}$ are null bonds.
12. All other edges have strength 1.

The $2k \times 2k$ Construction B snaked proofreading block is constructed as follows:

1. The west sides of tiles $T_{1,2i-1}$ are null bonds, where $i = 1, 2, \ldots, k-1$.
2. The north sides of tiles $T_{2i,1}$ are null bonds, where $i = 1, 2, \ldots, k-1$.
3. Edges on the north sides of tiles $T_{2i,2i+1}$ have strength 2, where $i = 1, 2, \ldots, k-1$.
4. Edges on the west sides of tiles $T_{2i,2i-1}$ have strength 2, where $i = 1, 2, \ldots, k-1$.
5. All other edges have strength 1.

In both the constructions described above, the edge labels internal to each $2k \times 2k$ block are unique to that block.

Figure 9 shows the three different snaked proofreading constructions for 6×6 blocks. Observe that the 6×6 block of Construction A (resp. Construction B) has the same bottom-right 4×4 (resp. 5×5) sub-block as the 6×6 block of the Chen-Goel construction. This observation generalizes naturally to the $2k \times 2k$ constructions, as may be verified by comparing the general descriptions of the new constructions given above to the corresponding description given by Chen and Goel (2005).

6 Conclusion

In this paper, we studied through simulations the effect of malformed (*i.e.*, imperfectly-fabricated) tiles on the performance of various tile sets producing the Sierpinski and Binary Counter patterns. A significant contribution of this paper is the development of two new snaked proofreading tile set constructions that provide examples of tile sets that are robust with respect to errors intrinsic to the assembly process, and also with respect to malformed tiles.

It should be pointed out that in this paper, we restricted our attention to proofreading tile sets in which the error-control mechanism consists of the introduction of redundancy into the edge labels of tiles. Alternative error suppression mechanisms have indeed been proposed in the literature (for example, Fujibayashi et al. 2008) and it would be of interest to study how malformed tiles affect these kinds of proofreading mechanisms.

Acknowledgements This work was supported in part by a Discovery Grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada. The final revision of this work was performed while the second author was on leave at the Electrical Communication Engineering Department of the Indian Institute of Science, Bangalore.

References

Chen H, Goel A (2005) Error Free Self-Assembly using Error Prone Tiles. In: DNA Computing 10, Lecture Notes in Computer Science, vol 3384. Springer, Berlin Heidelberg New York, pp 62–75

Fujibayashi K, Zhang DY, Winfree E, Murata S (2008) Error Suppression Mechanisms for DNA Tile Self-Assembly and their Simulation. Natural Computing (online). DOI 10.1007/s11047-008-9093-9

Meng Y (2009) Error-Resilient Tile Sets for DNA Self-Assembly. MSc Thesis, Queen's University, Kingston, Ontario

Reif JH, Sahu S, Yin P (2005) Compact Error-Resilient Computational DNA Tiling Assemblies. In: DNA Computing 10, Lecture Notes in Computer Science, vol 3384. Springer, Berlin Heidelberg New York, pp 293–307

Rothemund PWK, Papadakis N, Winfree E (2004) Algorithmic Self-Assembly of DNA Sierpinski Triangles. PLoS Biol 2:2041–2053

Soloveichik D, Winfree E (2006) Complexity of Compact Proofreading for Self-Assembled Patterns. In: DNA Computing 11, Lecture Notes in Computer Science, vol 3892. Springer, Berlin Heidelberg New York, pp 305–324

Wang H (1961) Proving Theorems by Pattern Recognition II. Bell Syst Tech J 40:1–42

Winfree E (1998a) Algorithmic Self-Assembly of DNA. PhD Dissertation, California Institute of Technology

Winfree E (1998b) Simulation of Computing by Self-Assembly. Caltech CSTR: 1998.22. California Institute of Technology

Winfree E, Bekbolatov R (2004) Proofreading Tile Sets: Error Correction for Algorithmic Self-Assembly. In: DNA Computing 9, Lecture Notes in Computer Science, vol 2943. Springer, Berlin Heidelberg New York, pp 126–144