



## UvA-DARE (Digital Academic Repository)

### From Molecules to Mathematics

Ellis-Monaghan, J.; Jonoska, N.

**DOI**

[10.1007/978-981-19-9891-1\\_11](https://doi.org/10.1007/978-981-19-9891-1_11)

**Publication date**

2023

**Document Version**

Final published version

**Published in**

Visions of DNA Nanotechnology at 40 for the Next 40

**License**

CC BY

[Link to publication](#)

**Citation for published version (APA):**

Ellis-Monaghan, J., & Jonoska, N. (2023). From Molecules to Mathematics. In N. Jonoska, & E. Winfree (Eds.), *Visions of DNA Nanotechnology at 40 for the Next 40: A Tribute to Nadrian C. Seeman* (pp. 189-206). (Natural Computing Series). Springer. [https://doi.org/10.1007/978-981-19-9891-1\\_11](https://doi.org/10.1007/978-981-19-9891-1_11)

**General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

**Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

# From Molecules to Mathematics



Joanna Ellis-Monaghan and Nataša Jonoska

**Abstract** To celebrate the 40th anniversary of bottom-up DNA nanotechnology we highlight the interaction of the field with mathematics. DNA self-assembly as a method to construct nanostructures gave impetus to an emerging branch of mathematics, called here ‘DNA mathematics’. DNA mathematics models and analyzes structures obtained as bottom-up assembly, as well as the process of self-assembly. Here we survey some of the new tools from DNA mathematics that can help advance the science of DNA self-assembly. The theory needed to develop these tools is now driving the field of mathematics in new and exciting directions. We describe some of these rich questions, focusing particularly on those related to knot theory, graph theory, and algebra.

## 1 Introduction

Seeman’s ground-breaking work in DNA self-assembly that initiated the field of DNA nanotechnology has had surprisingly broad impacts on many other fields. In particular it has impacted the field of nanotechnology more widely [1–4], the science of computing and molecular programming (e.g., [5–10] as well as articles in the proceedings of DNAX for over 26 years), bottom-up nano-assembly [11–19] including computer science and mathematics [20–28]. Advances in the sciences often drive the creation of completely new mathematical fields. The field of DNA nanotechnology is a prime example of this phenomenon. It has been steadily spawning a slew of new mathematical problems. These problems are now giving birth to a new field of mathematics that might be called ‘DNA-mathematics’.

We live in a world dominated by computers, mobile devices, advances in technology, and the internet. We have foodwebs, social networks, and contact tracing in

---

J. Ellis-Monaghan (✉)  
University of Amsterdam, Amsterdam, The Netherlands  
e-mail: [j.a.ellismonaghan@uva.nl](mailto:j.a.ellismonaghan@uva.nl)

N. Jonoska  
University of South Florida, Florida, USA  
e-mail: [jonoska@mail.usf.edu](mailto:jonoska@mail.usf.edu)

© The Author(s) 2023  
N. Jonoska and E. Winfree (eds.), *Visions of DNA Nanotechnology at 40 for the Next 40*,  
Natural Computing Series, [https://doi.org/10.1007/978-981-19-9891-1\\_11](https://doi.org/10.1007/978-981-19-9891-1_11)

epidemiological models. The mathematics of interconnections, in particular network and graph theory, became essential for better understanding of many modern life situations. For example, graph drawing tools led to effective computer chip layouts while random graphs are often used in modeling the worldwide web. We now see this same phenomenon of new mathematics emerging from questions about shapes and interconnections driven by nucleic acids structures in biology and DNA nanotechnology.

In his pioneering article on DNA self-assembly Ned Seeman proposed using DNA molecules, called branched junction molecules, to build complex nanostructures [29]. These molecules are shaped like starfish with anywhere from three to twelve arms and can attach to one another via sequences of complementary DNA bases at the ends of their arms (sticky ends), as though the starfish are holding hands. Thus, for example, four three-armed branched junction molecules can join together to form the outline of a tetrahedron.

In 2006, a major advance in DNA self-assembly was obtained by Rothemund's introduction of DNA origami [30], a method where a single-stranded DNA plasmid outlines a pre-designed shape and about 200–250 short strands of DNA complementary to different locations of the plasmid are used to assemble and secure the structure. Because of the potential applications of self-assembling DNA nanotechnology, especially in medicine, and also in nanoscale robotics, circuitry, and biosensors, hundreds of laboratories around the world today focus on it.

Many challenges arise in designing DNA molecules that are to self-assemble into a desired shape. While some of the challenges involve chemical processes, many others involve structural questions such as which arms of the branched junction molecules should attach to which other arms, or how to route the scaffolding strand and staples through the desired structure, so that the smaller molecules then self-assemble into exactly the desired larger shape. As the experimental advances evolve rapidly, mathematical foundations to address new and upcoming questions that arise from laboratory experiments are becoming more essential.

DNA self-assembly now involves novel mathematical approaches and tools that inform the design of the structure both to assemble the nanocomplex as well as to ease the analysis of the experimental results. It becomes particularly exciting, from a mathematical perspective, when these approaches diverge from the original stimulus to problems of intrinsic mathematical interest independent of the initial application, thus significantly expanding the scope of the mathematical investigations.

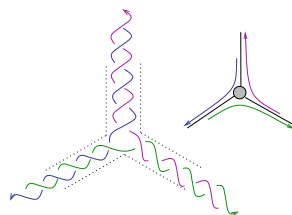
Fortunately, problems involving shapes and interconnections are at the heart of mathematics and mathematicians have become natural collaborators to DNA self-assembly researchers. New mathematical formalism is being developed to solve mathematical problems arising from DNA self-assembly. Many of the target molecular shapes have wireframe structures, such as the outlines of a cube [31] or octahedron [32], or 2D and 3D lattices [11, 14], or even triangular mesh bunny rabbits and lacy snowflakes [33]. Since the outlines of these wireframe structures correspond to the edges of a graph and the corners of the shape to the vertices of a graph, graph theory (in particular topological graph theory) and knot theory have emerged as an excellent platform to study assembly problems. However, existing graph and

knot theory lack sufficient descriptive properties to capture the essence of DNA self-assembly and cannot always address the new problems arising from the self-assembly application. Thus, a new subfield in mathematics is born, *DNA mathematics*, which develops new mathematical tools for, and arises from, bottom-up assembly.

As shared in the sections below, the mathematical problems driven by self-assembly processes are rich and prolific. They lead discrete mathematics and topology beyond current mainstream trends in these fields and hence break open new directions such as edge-outer embeddability, origami knotting, and new algebraic languages to describe structures. These theoretical directions are open-ended, generally scale independent, and will lay a broad foundation for future growth applicable to self-assembly in many settings, from nano to macro.

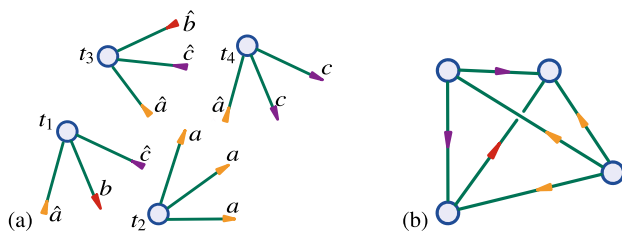
## 2 Flexible Tiles and New Graph Invariants

**Fig. 1** Three-armed branched DNA molecule seen as three-valent vertex in a graph with three half-edges. The single-stranded extensions of the arms encode the bond types



The appearances of the first three-dimensional DNA structures such as the cube [31], and the truncated octahedron [32], arrived at the same time as the idea of using nucleic acids and biomolecules for computations when molecular-based information processing was initiated with Adleman's seminal paper [34]. If computation is to be performed with molecular structures, assemblies of arbitrary 3D wireframe or graph-like molecules without inherent symmetry may be necessary. The first such construct of a graph with six vertices using branched junction molecules representing vertices, and regular duplex molecules representing edges, was reported in [35, 36]. These molecules contained non-paired nucleotides throughout the duplexes making the arms of the molecules flexible such that, in the process of assembly, their sticky ends could easily join their respective complements. After ligating the nicks (breaks in the strands), the resulting (graph) structure consisted of a single cyclic DNA strand, which could conform in at least two knotted topologies [36]. These experimental results initiated a mathematical model that captures some of the design challenges of the flexible armed tiles used in construction of spatially embedded graph structures.

A combinatorial abstraction that consists of a vertex with half-edges labeled by the sticky-end types on the arms of the branched junction molecule is called a *tile* (Fig. 1). It can be denoted as a multi-set of *bond types* indicating the types of sticky-end types flanking the ends of the arms. The complementary sticky-end types are



**Fig. 2** **a** Four tiles with their corresponding bonding types in colored arrowheads appearing at the end of the half-edges. The complementary bond types are indicated with reversed arrowheads. **b** A tetrahedron realized by a pot containing the tiles in **(a)**

denoted by marking bond types with two complementary versions, indicated with a symbol from an alphabet (e.g.,  $a, b, c, \dots$ ) and a hatted version of the symbol (e.g.,  $\hat{a}, \hat{b}, \hat{c}, \dots$ ). Multiple entries of the same bond type are indicated by the exponent to the corresponding symbol. For example, Fig. 2a shows tiles  $t_1 = \{\hat{a}, b, \hat{c}\}$ ,  $t_2 = \{a^3\}$ ,  $t_3 = \{\hat{a}, \hat{b}, \hat{c}\}$ , and  $t_4 = \{\hat{a}, c^2\}$ .

A collection of tile types forms a *pot*, and a target graph  $G$  (or other 3D wireframe structure) is *realized* by a pot if it can be obtained by matching the vertices of  $G$  with tile types and identifying two half-edges of tiles having complementary versions of the same bond type with an edge of the graph. In the most basic setting  $G$  is an abstract graph, but other models also consider the geometry of the target graph. Each pair of half-edges forming an edge is subject to the restriction that a symbol is always paired with its hatted version and vice versa. Abstractly, a graph is considered assembled from tiles, if every vertex of the graph corresponds to a vertex of a tile and each edge corresponds to a pair of half edges with complementary sticky-ends joined together to form a bond edge. This is equivalent to finding an edge-labeled orientation of the graph, with the arrows pointing from the unhatted to the hatted half edges making up the labeled edge. In theory, any covering of a graph with a cycle realized by a pot can also be realized by the pot, but entropy disfavors these larger constructs. Further description of the model can be found in [28, 37–39].

There are two aspects to consider in such a ‘pot’ and ‘tile’ set-up. One aspect considers properties of the pot: what types of graphs can be realized by a pot, how many isomorphism classes are there, are all structures that are realized complete (i.e., there are no unbonded half-edges), or, can they be always completed, etc. The other aspect asks questions about the graphs and the structures: what is the most ‘efficient’ pot (in terms of minimal number of tile types needed and bond types needed) that can realize a given graph, how are the properties of the pot related to other invariants of the graph in question, etc.

The properties of pots are most suitably studied with methods from linear algebra by associating a matrix to each pot whose  $ij$ -entries are ratios of bond type  $i$  for a tile type  $j$  present in a pot. Using this matrix, one can determine the stoichiometry of the self-assembly to ensure certain types of pot properties [39] (see also [37]). Although useful, this linear algebra approach does not capture the difficult task of

better understanding of the process of assembly. As larger complexes form in a test tube, the thermodynamic properties can change and further assembly of larger structures may depend of the entropic conditions in the test tube, hence an expansion of the model to include thermodynamic properties may be necessary. One can also associate computational complexity classes with types of pots, where the number of tiles used in a structure that computes a problem is a base for the associated classes [23]. It was shown that 3-colorability of a graph, or other NP-hard<sup>1</sup> problems could be solved in  $\mathcal{O}(1)$  bio-steps [23, 36] by physically assembling the structure that emulates the solution of the problem. This was also shown experimentally by Seeman's lab [7]. However, it seems we are yet to develop a good model that can encapsulate information processing by 3D structures, or a model that can tell us about computations by shapes.

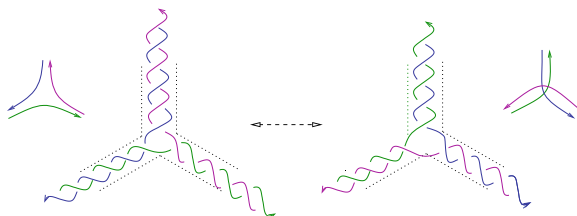
Determining an 'efficient' pot for a target graph  $G$  means specifying a pot of tiles that realizes combinatorially, with a minimum number of tile types, and using a minimum number of bond-edge types. Recall that in the assembly process there are typically  $\sim 10^{15}$  tiles of each type present in a tube, each tile can be used within a structure multiple times (or often taken, an arbitrarily large number of times). For the construction of  $G$  one can require an edge labeled orientation of the graph using a minimal alphabet for the labels. This becomes a new graph invariant  $B(G)$ , the *minimal bond alphabet*. In order to prevent the self-aggregation of tiles during initial synthesis, a natural experimental requirement can be added so that no two half-edges of the same tile have complementary labels. This corresponds to prohibiting loops in the target graph. One can further consider the *minimal set of vertex types* (the resulting molecular components) needed for construction of the graph  $G$ , an invariant  $T(G)$ . The invariants  $T(G)$  and  $B(G)$  are new graph invariants of intrinsic interest that have yet to be determined for most graphs. Except for a few special graph classes in some specific pot setting, these combinatorial questions are wide open. Familiar graph theoretical tools such as coloring, chromatic numbers, classical graph automorphism, etc., do not appear to determine  $T(G)$  or  $B(G)$  in any of the settings. The chromatic number provides a lower bound in some settings, although a poor one as it and  $T(G)$  can be arbitrarily far apart [28].

The problem can further be confounded by the constraints of several different experimental settings, including the degree of flexibility or rigidity of the arms and the strength of the cohesive sites, as well as yield considerations such as whether or not the incidental creation of complete complexes smaller than the target graph is acceptable. In [40] it was confirmed that it is NP-hard to determine the output of a pot and to determine whether a pot that assembles a target structure will also assemble unwanted smaller structures.

Mathematical formalism, models, and design tools for types of self-assembly can be found here [28, 37, 38, 40, 41], with [42] further specifying the inter-arm angles and cohesive end rotation orientations for rigid tiles. These resources provide provably optimal design strategies, i.e., combinatorial specification for the minimum

---

<sup>1</sup> Loosely speaking, NP-hard problems are those shown to be at least as difficult as a large set of other problems for which no fast algorithms are known.



**Fig. 3** Changing strand connections at a three-valent vertex in a graph. As DNA strands have polarity with a phosphate end (5') and a hydroxyl group on the other end (3'), the arrowheads indicate the 3' ends. The strands bind with opposite polarities. A three-armed junction molecule can exhibit two types of connections as depicted, e.g., the green strand has a direction left to right in the left figure, while it has direction left to top in the right figure

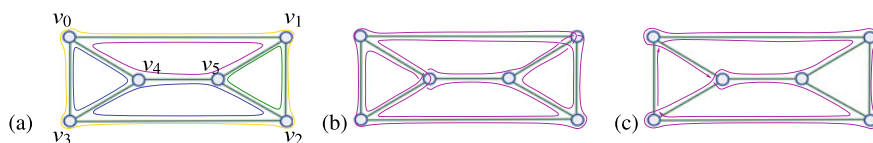
number of cohesive regions and the minimum number of branched junction molecule types, to assembly a given wireframe target structure, for several common graph classes. Further computations of these invariants for specific graphs such as Platonic and Archimedes solids and prisms can be found in [43]. A further extensive body of work can be found on DNA Wang tiles, which are the restrictive case of completely rigid four-armed tiles. See, among others, Refs. [5, 44–49].

### 3 DNA Strand Routing and Topological Graph Theory

Determining routes for a scaffolding strand throughout assembly targets is integral to both DNA origami [27, 50, 51] and experimental verification of graph constructs [7]. A vertex of a graph structure can be traced by DNA in multiple ways such that the resulting DNA structure is assembled by different sets of cross-hybridizing cyclic molecules.

For a three-valent vertex, the DNA structure representing such vertex can have two local strand configurations (vertex connections) as shown in Fig. 3. One configuration can be represented by ‘non-crossing’ strands at a vertex, and the other is obtained by ‘crossing strands’. By changing vertex connections, the number of cyclic molecules assembling the structure can vary. Figure 4a and b show two strand outlines of the same triangular prism graph. In Fig. 4a, the strand connections at each vertex in this planar representation do not ‘cross’ each other and the graph is outlined by five cyclic strands. By changing the strand connection at vertices  $v_1$  and  $v_4$  with the other configuration, the graph can be routed by a single circuit, hence outlined by a single cyclic molecule (as shown in Fig. 4b).

Early work focused on number of cyclic molecules in such double-strand covers, where the target graph is covered by a set of circuits so that every edge is covered exactly twice, as in Fig. 4a [52]. These circuits form the facial walks of an oriented embedding of the graph, and thus correspond to a strong version of the cycle double cover conjecture, as described below. Double-strand covers have re-emerged recently in novel origami methods such as [53].



**Fig. 4** Possible routes for DNA strands in a triangular prism graph: **a** following the faces of a plane embedding produces 5 faces, corresponding to a maximum strand number of 5. **b** by changing strand connections at two vertices, the graph is traced by a single strand, reflecting that it is upper-embeddable on a double torus (see Fig. 5a). **c** A minimal length reporter strand for the graph, giving an edge-outer embedding on the torus (see Fig. 5b)

In DNA settings, strand routings of structures require turns at vertices that are constrained, i.e., routes may not doubleback on an edge. Because DNA strands are oriented, any repeated edges in a strand route must be traversed in the opposite direction when revisited. Thus, by folding a DNA origami in a graph structure, the strand routing objective is to find a route in the graph which meets these constraints, traverses every edge, ideally, with a minimum number of repeated edges. A similar problem arises in determining a route for the reporter strand, where after an experiment has been conducted, a single strand traversing the structure is extracted from the assembled construct and analyzed to yield the experimental data (see [7]). Here again, the objective is a route covering the graph with a minimum number of repeated edges. Such a minimal strand routing for the triangular prism graph is shown in Fig. 4c. In this case, the connections at three vertices,  $v_0$ ,  $v_3$ , and  $v_4$  are changed relative to the connections in Fig. 4a.

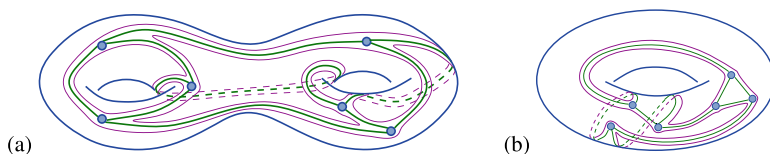
The strand routing problem is in general intractable, even in the special case of an Eulerian surface mesh when an optimal route follows face boundaries. In this case the problem corresponds to finding  $A$ -trails in the surface, which are Eulerian circuits that turn either left or right at each vertex. This problem is known to be NP-hard (see [54]) even on the plane. In [51] it is proven that the strand routing problem remains NP-hard even for graphs of maximum degree 8. A survey of approaches to the strand routing problem is given in [27]. The problem can be translated to the traveling salesman problem (TSP), which makes the wealth of software available for solving the TSP available to DNA origami applications.

A notable alternative approach is given in [33]. Provided that the graph is an augmented triangulation of a surface topologically equivalent to a sphere, there is a fast algorithm for routing the scaffolding strand, albeit at the cost of duplicating a number of edges in the final nanostructure.

Mathematically, different routings correspond to different cellular embeddings of a graph in a surface (drawings of a graph in a surface such that its edges do not cross each other and such that the complement of the graph in the surface is a set of topological disks). In general, the number of strands outlining a graph is counter-proportional to the genus<sup>2</sup> of the surface in which the graph is embedded.

<sup>2</sup> Intuitively, the genus of an orientable surface is the number of holes, or handles it has, so that a torus has genus one, a double torus has genus two, and so forth.





**Fig. 5** Embeddings of the triangular prism graph in Fig. 4. **a** Embedding of the graph in a double torus. The graph can be realized as a tertiary structure of a single DNA strand, shown here tracing the single face. **b** An embedding of a graph in a torus with a minimum length reporter strand tracing face boundary that covers all edges at least once

As has repeatedly been the case with self-assembly applications, a wealth of new mathematical directions have emerged from the design objective, with *edge-outer embeddability* as described below being a particularly rich example expanding on topological graph theory.

An *edge-outer embedding* is a cellular embedding of a graph in an orientable surface where every edge lies on a special ‘outer’ face (there may be other faces too). Figure 5 shows an edge-outer embedding of the triangular prism graph in the double torus and the torus corresponding to the reporter strand routings in Fig. 4b, c. In [55] and [56] it was shown that the facial walk of an edge-outer face in an edge-outer embedding of the target graph exactly captures reporter strand routing constraints. While outer-planar and outer-projective-planar graphs, i.e., graphs in the plane or projective plane with all vertices on a single distinguished face, have been heavily studied [57], edge-outer embeddable graphs are an entirely new, yet very natural, construct.

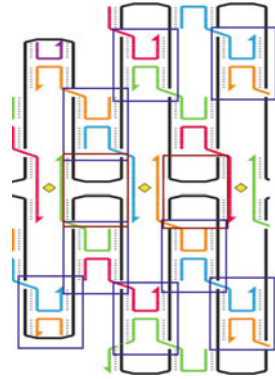
Conventional graph theoretical tools do not apply directly to edge-outer embeddability. For example, the Chinese postman problem lacks the bidirectional constraint while upper embeddability results require every edge to be covered exactly twice [58]. That every graph has such a reporter strand route, and hence an edge-outer embedding was shown in [55], with a short algorithmic proof given in [56]. At the heart of the proof is the reconfiguration shown in Fig. 3, which changes the cyclic order of the edges about a vertex. Finding a minimum such route, that is an embedding with a smallest possible edge-outer face, is in general NP-hard [56].

Such a computational complexity observation introduces a wealth of further open questions. If the graph is Eulerian, an optimal reporter strand route is simply an Euler circuit. Thus, in this case, Fleury’s algorithm provides a polynomial time solution to the problem. For what classes, other than Eulerian graphs, are there polynomial-time algorithms to find optimal reporter strand routes? Does there exist a polynomial-time algorithm guaranteed to return a route that is within  $x\%$  of the optimal length for some reasonable  $x$ ?

It is also natural, both mathematically and for the application, to consider *maximum* length routes. If the graph is upper embeddable, i.e., has an orientable one-face embedding, then this embedding is a maximum edge-outer embedding with every edge covered twice in the facial walk (see Fig. 4b, c); however, not every graph is upper embeddable. As with the dichotomy in the complexity of maximum and minimum genus of the graph, it is possible that the maximum length problem is tractable.

### 4 DNA Origami and New Algebraic Structures

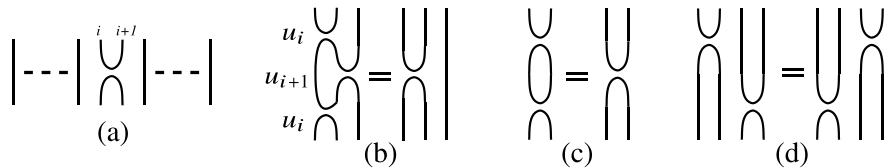
**Fig. 6** A segment of a DNA origami structure with black scaffold and colored staples. The generators of the proposed monoid are boxed



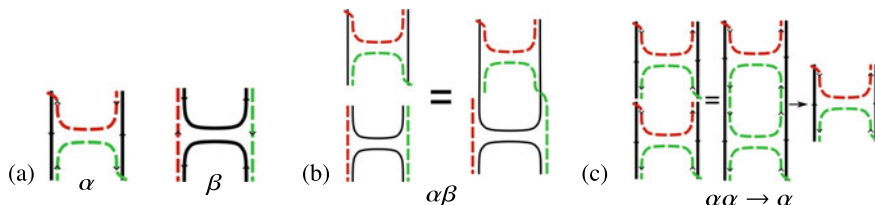
Since their introduction to the scientific community, DNA origami structures [59] have become one of the most prevalent experimental substrates for a variety of 3D structures [12]. The method uses a single-stranded DNA plasmid vector as a scaffold that outlines the shape of the desired structure with help of so-called staple strands. The staple strands are short segments of single-stranded DNA (about 24 to 32 nucleotides long) that span across two or three segments of the scaffold strand to bind it in place. In order to achieve stability of the construct, it is necessary that the staple strands cross each other within the structure in an antiparallel way. An example of a portion of an origami structure is depicted in Fig. 6. (Figure adjusted from [30] with added boxes). Systematic mathematical methods to describe DNA origami structures have not been established. An algebraic language that describes DNA origami motivated by the Temperley-Lieb algebra that has been extensively used in physics and knot theory, particularly with the Jones polynomial and the Kauffman bracket, was introduced in [60, 61]. Such languages could provide a method for modifying (through strand displacements) a given design to achieve either a more effective and stable structure or a completely new geometric shape.

A well-studied Jones monoid  $\mathcal{J}_n$  has  $n$  generators are  $u_1, \dots, u_n$ . Each generator can be represented with  $n + 1$  lines such that  $u_i$  has cap/cup connection of lines  $i$  and  $i + 1$  as shown in Fig. 7a while all other lines are vertical. The product of generators is diagrammatically represented by concatenating the corresponding diagrams vertically. Figure 7b shows the product  $u_i u_{i+1} u_i$  to the left of the equality where the vertical lines have not been depicted.

The set of relations of  $\mathcal{J}_n$  is depicted in Fig. 7b–d. For example, (b) represents the relation  $u_i u_{i+1} u_i = u_i$ . Although the generators and relations are symbolically written, they correspond directly to the depicted diagrams and planar isotopy. Taking this diagrammatic usage as an advantage, we use a monoid version that is a generalization of  $\mathcal{J}_n$  for describing DNA origami.



**Fig. 7** Generator  $u_i$  in **(a)** and three types of relations in **(b, c, d)** of the Jones monoid. **b** Conjugacy relation  $u_i u_{i+1} u_i = u_i$ , **c** idempotent relation  $u_i u_i = u_i$ , **d** commuting relation  $u_i u_j = u_j u_i$  for  $|i - j| \geq 2$



**Fig. 8** **a** Two types of generators, cross-over staples connecting straight-line portions of the scaffold (denoted with  $\alpha$ ) and straight staples connecting two cross-over turning tips of the scaffold (denoted with  $\beta$ ). **b** Multiplying generators is represented by structures connecting scaffold segments and respective staples, when they don't ‘cross-over’ the scaffold strand. **c** An idempotent rule; the product  $\alpha\alpha$  has the top and bottom structure the same as  $\alpha$ . The cyclic portion in the middle does not affect further products

The basis of the origami algebraic language is a monoid  $\mathcal{O}_n$  inspired by  $\mathcal{J}_n$ . It is defined by generators and relations whose diagrammatic representations and closures are parts of DNA origami with  $n + 1$  scaffold strands tracing the structure (e.g., in Fig. 6 the structure is traced with 6 passings of the scaffold across top to bottom, and one can consider the structure as a representative of  $\mathcal{O}_5$ ). The monoid has two types of generators, those that correspond to the scaffold strands connecting straight vertically in the structure ( $\alpha$ 's) and those that correspond to staple strands connecting across ( $\beta$ 's). The generators  $\alpha_i$  and  $\beta_i$  of the origami monoid  $\mathcal{O}_n$  of  $n$  strands are depicted in Fig. 8a, where the cap and cup are placed at the  $i$ -th position from the left, and the vertical lines surrounding these positions are not depicted.

We define a DNA origami monoid  $\mathcal{O}_n$  on  $n + 1$  strands by generators and relations as follows. Generators of  $\mathcal{O}_n$  are  $\alpha_i$ , for  $i = 1, \dots, n$ . The  $\alpha_i$ 's represent local DNA foldings of a pair of staples of the form of cap and cup as indicated on the left in Fig. 8a. An additional set of generators  $\beta_i$  corresponds to pairs of caps and cups of the scaffold strand as indicated in the figure to the right. The index  $i$  indicates that the caps and cups are between the  $i$ th and  $i + 1$ st strands of the structure. The set of relations is defined analogously to those of  $\mathcal{J}_n$ , while the closure of the diagram is defined in a manner similar to the so-called plat closure in knot theory [62].

To justify modeling DNA origami structures by words over the generators we make a correspondence between concatenations of generators  $\alpha_i$ ,  $\beta_i$  and connections of DNA segments. For a natural number  $n \geq 2$ , the set of generators of the monoid  $\mathcal{O}_n$

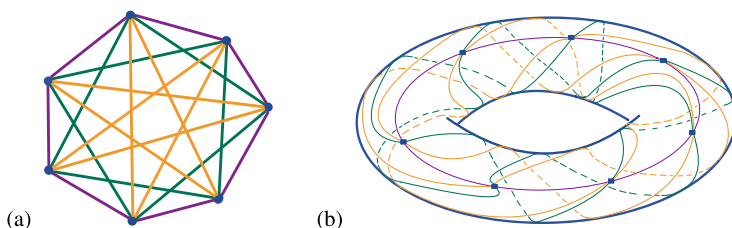
is the set  $\Sigma_n = \{\alpha_1, \alpha_2, \dots, \alpha_n, \beta_1, \beta_2, \dots, \beta_n\}$ . For a product of two generators  $x_i$  and  $y_j$  in  $\Sigma_n$ , we place the diagram of the first generator above the second, lining up the scaffold strings of the two generators, and then we connect the respective scaffold strands. If the two generators are ‘far’ apart, that is, if for indices  $i$  and  $j$  we have  $|i - j| \geq 2$ , then no staple connection is performed. If the two generators are adjacent, that is, if for indices  $i$  and  $j$  it holds that  $|i - j| \leq 1$ , then we connect part of the staples. Since the staples are not too long within an origami structure, spanning only two or three segments of the scaffold, a convention of connecting staples representing a product of generators is motivated by the manner in which staples connect within the DNA origami structure. The staples of  $\alpha$ -type generators protrude “outside” of the scaffold in Fig. 8a. The staples are connected everywhere *except* when two non-extending staple-ends would have to cross a scaffold to connect. With this convention it means that  $\alpha_i\beta_i$  and  $\beta_i\alpha_i$  are two distinct words, i.e., that  $\alpha$ s and  $\beta$ s do not freely commute. The rules of connecting scaffold strands and staples assures that concatenation of three or more generators is associative. The graphical representation of a product  $\alpha_i\beta_i$  is shown in Fig. 8b where the vertical strands surrounding the generators are not depicted.

The generators of the origami monoid satisfy a set of relations, according to the types of graphical structures they represent. One of the relations is a generator idempotent relation as shown in Fig. 8c. The relations are extensions of the relations of the Jones monoids. In [61] two scenarios of relations and graphical representations of the elements of the corresponding monoids are given. For each scenario, the number of all possible structures is provided through the number of equivalence classes of words (or elements in the corresponding origami monoid). Also a polynomial time algorithm exists that computes the shortest word for each equivalence class. A connection between the Green’s relations of an origami monoid and those of a direct product of Jones monoids [62] is given in [60]. In particular, it was shown that an epimorphism  $p : \mathcal{O}_n \rightarrow \mathcal{J}_n \times \mathcal{J}_n$  induces the bijective correspondence on Green’s classes.

The definition of origami monoids is motivated by the Jones monoids, and the origami structure, concatenating two types of strands (scaffolds and staples), implies two types of generators with similar relations. This construction of doubling generators and imposing substitution relations can be generalized to other algebraic structures, or it can be generalized with more than two types of generators. These are algebraic problems directly arising from the experimental design of DNA origami.

## 5 DNA Origami and Origami Knots

DNA origami assembly can be confounded by knotting in the scaffolding strand. For example, in a preliminary experiment, an essentially planar target did not form well when a simply knotted (trefoil) scaffold was used [63]. Thus, tools are needed to avoid inadvertently knotted routes when designing the increasingly sophisticated targets of DNA origami. On the other hand, Seeman has shown it is possible to engineer single-



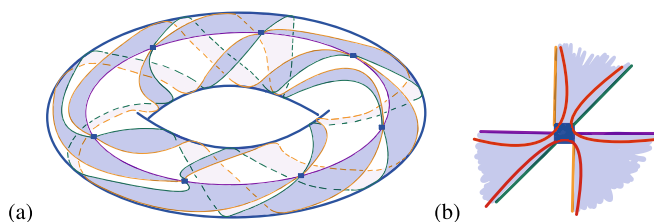
**Fig. 9** **a** A complete graph with seven vertices,  $K_7$ , with three Hamiltonian cycles colored distinctly. **b** An embedding of  $K_7$  on a torus. The edges of the yellow and green cycles go around the torus handle. Both those cycles are knotted

stranded DNA with specified knotted topologies [64], and this capacity for controlled knotting may be used intentionally to design better (higher yield, larger, smaller, more symmetric, more robust, more topologically complex, etc.) nanostructures by deliberately exploiting the topology of the knotted scaffold.

When routing a single scaffold through a graph-like target structure, a typical design constraint is avoiding self-crossings [24, 33]. If the target structure is modeled by an Eulerian graph cellularly embedded on an oriented surface in 3-space, then some of these routes correspond to *A-trails*, which are Eulerian circuits that turn either ‘left’ or ‘right’ at each vertex. If the surface is a sphere, then all *A-trails* in the target structure are necessarily unknotted, but for higher-genus surfaces there are settings in which every *A-trail* is knotted [26, 27]. The complete graph on seven vertices,  $K_7$ , is shown in Fig. 9a, and an embedding of  $K_7$  on a torus is shown in Fig. 9b. Every vertex of  $K_7$  has even valency (i.e., valency 6) and hence, the graph is Eulerian. Three distinct Hamiltonian cycles are depicted, purple, green and yellow. The embeddings of the yellow and the green cycles are knotted.

Since standard DNA origami scaffold strands are unknotted, the problems arise of determining whether there exists an unknotted route for a scaffolding strand in a given geometrically (or surface) embedded target graphs, and of characterizing graphs which have unknotted routes. Once again, a new area of mathematics emerges, as determining knotted and unknotted routing trails is fundamentally different from previously studied knots and links in graphs. Prior work focused on intrinsically knotted and linked cycles in graphs (see, for example, [65–67]), appearing as cycles in the graph. However, for this application, the knots are Eulerian circuits rather than cycles. For example, Conway and Gordon [67] showed that every embedding of  $K_7$  in  $\mathbb{R}^3$  has at least one knotted cycle. Figure 9 shows an embedding of  $K_7$  on the torus. By Conway and Gordon [67], it contains at least one knotted Hamilton cycle (the green cycle, for example).

However, in contrast, [24, 26] have shown that every *A-trail* in the embedding of  $K_7$  in the torus is unknotted. This follows since the embedding of  $K_7$  in the torus is checkerboard colorable, and hence every *A-trail* bounds the union of the black regions, and hence a disk. It is consequently unknotted (Fig. 10). Moreover, in [24]



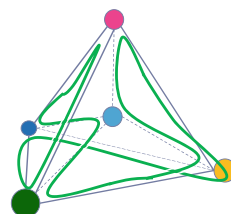
**Fig. 10** **a** A checker-board coloring of the embedded  $K_7$ , **b** A-trail transitions at a vertex of the embedding joining the shaded regions of the embedded  $K_7$ . These A-trails bound a disk consisting of the shaded regions on the torus and therefore they outline  $K_7$  with an unknotted routing

it is shown that every Eulerian circuit is knotted on a torus if and only if there is a non-checkerboard colorable embedding of the graph.

For geometrically embedded graphs that are not necessarily embedded on some surface, and hence for which A-trails are not necessarily defined, the initial challenge is formalizing the constraints on how the scaffolding strand may pass through vertices.

A-trails can be generalized to *O-trails*, which permits analyzing the knotting of strand routes in more general geometric embeddings of graphs in space, or embedding on a surface where ‘non-crossing’ smoothings at the vertices are defined [24]. For example, Fig. 11 shows  $K_5$  embedded as a tetrahedron with a body-center vertex, together with an *O-trail* through it.

**Fig. 11** An *O-trail* in a geometric embedding of  $K_5$



*O-trails* are Eulerian circuits in geometrically embedded Eulerian graphs so that at each vertex the edge pairings determined by the circuit are all non-crossing (i.e., there is a topological disk containing the half edges about the vertex in which both the turnings determined by the circuit and the orthogonal turnings are non-crossing). For A-trails in a surface mesh, the disk is just a small neighborhood of the vertex, so A-trails are a special subclass of *O-trails*.

The DNA origami application needs to understand and control the behavior of *O-trails*, both knotted and unknotted, in fixed geometric graphs, as well as over all possible embeddings of an abstract graph. Finding *O-trails* is NP-complete in general, since finding A-trails is, even for plane graphs, and the problem of determining if a knot is unknotted is NP-hard.

Both unknotted  $O$ -trails and constrained knot embeddings are entirely new directions in knot theory. There has been extensive prior work on knots in graphs, but this focused on intrinsically linked cycles or knotted Hamiltonian cycles [66–68]. Here however, the DNA origami constraints lead to questions of knots in Eulerian circuits, which opens an analogous and equally rich line of inquiry, but now in a completely new setting. Although embedding a given knot on a standard surface is known, considering geometric knot invariants under requirements imposed by DNA conformation is novel. Furthermore, characterizing graphs with unknotted  $O$ -trails will help identify target structures amenable to origami self-assembly from unknotted scaffolding strands and complements current experimentation [15]. In most applications a specific geometric embedding of the target is sought, so a pragmatic goal is to characterize classes of embedded graphs for which determining unknotted  $O$ -trails is computationally tractable.

## 6 Where Next?

The mathematical models discussed here generally follow what is becoming a common pattern. The first step is always developing a theoretical formalism that simultaneously captures the essence of a design problem, while providing a foundation for the following theoretical work. The second step is developing preliminary results for specific experiments.

When moving from specific experiments to seeking general strategies, e.g., via fast computer algorithms, particular design problems often can be prohibitively difficult. Thus the third step frequently is providing fast algorithms or proving that fast algorithms for general solutions might not be possible, i.e., are NP-hard.

Discovering that a DNA self-assembly problem is NP-hard is exciting theoretically, because it immediately opens a plethora of related computational problems. These include seeking approximation algorithms, finding optimal solutions for particular families of problems, and devising pragmatic approaches for urgently needed special cases.

Although an NP-hardness result does not help a lab trying to conduct its next experiment, it does prevent wasted effort seeking general strategies. Furthermore, a specific NP-hard problem may be reduced to another known NP-hard problem, for example, the Traveling Salesman Problem (TSP), but for which there already exist robust tools such as fast approximation algorithms and algorithms optimized for special cases. These then may be adapted to the self-assembly problem, as with the TSP reduction for strand routing in [27].

DNA self-assembly is now a rich source of theoretical problems. These problems and their solutions advance both the mathematics and the self-assembly processes. The lab constraints lead to new mathematics and often breakthrough new directions, as we point out here. The mathematical foundations supporting innovations in science and industry are often seen years after they are developed, and are sometimes unacknowledged, because by the time the theory has been applied, it has entered

the mainstream of engineering. This emergent mathematical theory, this new ‘DNA mathematics’, lays the foundations that can support the further growth of DNA self-assembly technologies, and whose potential applications, although inspired by DNA self-assembly, may still be yet to come.

**Acknowledgements** The authors dedicate this paper to the memory of Nadrian C. Seeman in gratitude for longlasting collaboration and for his original and insightful thinking accompanied by a plethora of experimental results that initiated DNA mathematics. We also thank the E. Czeizler, H.J. Hoogeboom and G. Pangborn for their thoughtful suggestions. This research was supported by NSF grants CCF-2107267, DMS-1800443, and DMS-2054321, as well as the NSF-Simons Southeast Center for Mathematics and Biology (SCMB) through the grants National Science Foundation DMS1764406 and Simons Foundation/SFARI 594594.

## References

1. Y. Kamiya, H. Asanuma, Light-driven DNA nanomachine with a photoresponsive molecular engine. *Acc. Chem. Res.* **47**(6), 1663–1672 (2014)
2. S. Jia, J. Wang, M. Xie, J. Sun, H. Liu, Y. Zhang, J. Chao, J. Li, L. Wang, J. Lin, K.V. Gothelf, C. Fan, Programming DNA origami patterning with non-canonical DNA-based metallization reactions. *Nat. Commun.* **10**(1), 5597 (2019)
3. C. Cui, D.H. Park, D.J. Ahn, Organic semiconductor-DNA hybrid assemblies. *Adv. Mater.* **32**(51), 2002213 (2020)
4. Y. Li, L. Song, B. Wang, J. He, Y. Li, Z. Deng, C. Mao, Universal pH-responsive and metal-ion-free self-assembly of DNA nanostructures. *Angew. Chem. Int. Ed.* **130**(23), 6892–6895 (2018)
5. D. Woods, D. Doty, C. Myhrvold, J. Hui, F. Zhou, P. Yin, E. Winfree, Diverse and robust molecular algorithms using reprogrammable DNA self-assembly. *Nature* **567**(7748), 366–372 (2019)
6. P.W.K. Rothmund, N. Papadakis, E. Winfree, Algorithmic self-assembly of DNA sierpinski triangles. *PLoS Biol.* **2**(12), e424 (2004)
7. G. Wu, N. Jonoska, N.C. Seeman, Construction of a DNA nano-object directly demonstrates computation. *Biosystems* **98**(2), 80–84 (2009)
8. B. Chakraborty, N. Jonoska, N.C. Seeman, A programmable transducer self-assembled from DNA. *Chem. Sci.* **3**(1), 168–176 (2012)
9. R.D. Barish, P.W.K. Rothmund, E. Winfree, Two computational primitives for algorithmic self-assembly: copying and counting. *NanoLetters* **5**, 2586–2592 (2005)
10. H. Gu, J. Chao, S.-J. Xiao, N.C. Seeman, A proximity-based programmable DNA nanoscale assembly line. *Nature* **465**, 202–205 (2010)
11. E. Winfree, F. Liu, L.A. Wenzler, N.C. Seeman, Design and self-assembly of two-dimensional DNA crystals. *Nature* **394**(6693), 539–544 (1998)
12. Y. Hao, M. Kristiansen, R. Sha, J.J. Birktoft, C. Hernandez, C. Mao, N.C. Seeman, A device that operates within a self-assembled 3D DNA crystal. *Nat. Chem.* **9**(8), 824–827 (2017)
13. X. Wang, R. Sha, M. Kristiansen, C. Hernandez, Y. Hao, C. Mao, J.W. Canary, N.C. Seeman, An organic semiconductor organized into 3D DNA arrays by “bottom-up” rational design. *Angew. Chem. Int. Ed.* **56**(23), 6445–6448 (2017)
14. J. Zheng, J.J. Birktoft, Y. Chen, T. Wang, R. Sha, P.E. Constantinou, S.L. Ginell, C. Mao, N.C. Seeman, From molecular to macroscopic via the rational design of a self-assembled 3D DNA crystal. *Nature* **461**(7260), 74–77 (2009)
15. F. Hong, F. Zhang, Y. Liu, H. Yan, DNA origami: Scaffolds for creating higher order structures. *Chem. Rev.* **117**(20), 12584–12640 (2017)



16. R. Inuma, Y. Ke, R. Jungmann, T. Schlichthaerle, J.B. Woehrstein, P. Yin, Polyhedra self-assembled from DNA tripods and characterized with 3d DNA-PAINT. *Science* **344**(6179), 65–69 (2014)
17. S. Douglas, A. Marblestone, S. Teerapittayanon, A. Vazquez, G. Church, W. Shih, Rapid prototyping of 3d DNA-origami shapes with caDNAo. *Nucleic Acids Res.* **15**, 5001–5006 (2009)
18. W. Xu, B. Nathwani, C. Lin, J. Wang, E. Karatekin, F. Pincet, W. Shih, J.E. Rothman, A programmable DNA origami platform to organize snares for membrane fusion. *J. Am. Chem. Soc.* **138**(13), 4439–4447 (2016). (PMID: 26938705)
19. D. Liu, Y. Shao, G. Chen, Y.C. Tse-Dinh, J.A. Piccirilli, Y. Weizmann, Synthesizing topological structures containing RNA. *Nat. Commun.* **8**(1), 14936 (2017)
20. E. Winfree, *Algorithmic Self-Assembly of DNA*. Ph.D thesis (California Institute of Technology, 1998)
21. D. Woods, Intrinsic universality and the computational power of self-assembly. *Philos. Trans. R. Soc. A. Math. Phys. Eng. Sci.* **373**(2046), 20140214 (2015)
22. M. Vasić, D. Soloveichik, S. Khurshid, Crn++: Molecular programming language. *Nat. Comput.* **19**(2), 391–407 (2020)
23. N. Jonoska, G.L. McColm, Complexity classes for self-assembling flexible tiles. *Theoret. Comput. Sci.* **410**(4–5):332–346 (2009). Computational Paradigms from Nature
24. A. Mohammed, N. Jonoska, M. Saito, The topology of Scaffold routings on non-spherical mesh wireframes, in *26th International Conference on DNA Computing and Molecular Programming (DNA 26)*. ed. by C. Geary, M. J. Patitz, vol. 174 of *Leibniz International Proceedings in Informatics (LIPIcs)*, (Dagstuhl, Germany), pp. 1:1–17 (Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020)
25. M.N. Ellingham, J.A. Ellis-Monaghan, Edge-outer graph embedding and the complexity of the DNA reporter strand problem. *Theoret. Comput. Sci.* **785**, 117–127 (2019)
26. A. Morse, W. Adkisson, J. Greene, D. Perry, B. Smith, J. Ellis-Monaghan, G. Pangborn, DNA origami and unknotted A-trails in torus graphs. *J. Knot Theor. Ramifications* **29**(07), 2050041 (2020)
27. J. Ellis-Monaghan, G. Pangborn, N. Seeman, S. Blakeley, C. Disher, M. Falcigno, B. Healy, A. Morse, M. Singh, B. Westland, Design tools for reporter strands and DNA origami scaffold strands. *Theor. Comput. Sci.* **671**(6), 69–78 (2017)
28. J. Ellis-Monaghan, G. Pangborn, L. Beaudin, D. Miller, N. Bruno, A. Hashimoto, Minimal tile and bond-edge types for self-assembling DNA graphs, in *Discrete and Topological Models in Molecular Biology*, ed. by N. Jonoska, M. Saito, pp. 241–270 (Springer, Berlin/Heidelberg, 2014)
29. N.C. Seeman, Nucleic acid junctions and lattices. *J. Theoret. Biol.* **99**(2), 237–47 (1982)
30. P.W. Rothmund, Folding DNA to create nanoscale shapes and patterns. *Nature* **440**(7082), 297–302 (2006)
31. J. Chen, N. Seeman, Synthesis from DNA of a molecule with the connectivity of a cube. *Nature* **350**, 631–633 (1991)
32. Y. Zhang, N. Seeman, The construction of a DNA truncated octahedron. *J. Am. Chem. Soc.* **160**, 1661–1669 (1994)
33. E. Benson, A. Mohammed, J. Gardell, S. Masich, E. Czeizler, P. Orponen, B. Högberg, DNA rendering of polyhedral meshes at the nanoscale. *Nature* **523**(7561), 441–444 (2015)
34. L. Adleman, Molecular computation of solutions to combinatorial problems. *Science* **266**(5187), 1021–1024 (1994)
35. N. Jonoska, P. Sa-Ardyen, N. Seeman, Computation by self-assembly of DNA graphs. *Genet. Program. Evolvable Mach.* **4**, 123–137 (2003)
36. P. Sa-Ardyen, N. Jonoska, N. Seeman, Self-assembling DNA graphs. *Nat. Comput.* **4**, 427–438 (2003)
37. J. Ellis-Monaghan, N. Jonoska, G. Pangborn, Tile-based DNA nanostructures: mathematical design and problem encoding, in *Algebraic and Combinatorial Computational Biology*, ed. by R. Robeva, M. Macauley, MSE/Mathematics in Science and Engineering, pp. 35–60 (Academic Press, 2019)

38. S. Bonvicini, M.M. Ferrari, On the minimum number of bond-edge types and tile types: an approach by edge-colorings of graphs. *Discrete Appl. Math.* **277**, 1–13 (2020)
39. N. Jonoska, G.L. McColm, A. Staninska, On stoichiometry for the assembly of flexible tile DNA complexes. *Nat. Comput.* **10**(3), 1121–1141 (2011)
40. L. Almodóvar, J. Ellis-Monaghan, A. Harsy, C. Johnson, J. Sorrells, *Computational Complexity and Pragmatic Solutions for Flexible Tile Based DNA Self-Assembly* (2021)
41. J. Ellis-Monaghan, G. Pangborn, L. Beaudin, D. Miller, N. Bruno, A. Hashimoto, Minimal tile and bond-edge types for self-assembling DNA graphs, in *Discrete and Topological Models in Molecular Biology, Natural Computing Series* (2013)
42. M. Ferrari, A. Cook, A. Houlihan, R. Rouleau, N. Seeman, G. Pangborn, J. Ellis-Monaghan, Design formalism for dna self-assembly of polyhedral skeletons using rigid tiles. *J. Math. Chem.* **56**, 1365–1392 (2018). Funding Information: Acknowledgements The work of Joanna Ellis-Monaghan, Greta Pangborn, and Nadrian C. Seeman was supported by the National Science Foundation (NSF) under Grant DMS-1332411. Publisher Copyright: 2018, Springer International Publishing AG, part of Springer Nature
43. Ellis-Monaghan and Pangborn DNA Self-Assembly Website. <http://sites.google.com/site/nanoselfassembly>. Accessed 14 Aug 2017
44. E.D. Demaine, M.J. Patitz, R.T. Schweller, S.M. Summers, Self-assembly of arbitrary shapes using RNase enzymes: meeting the Kolmogorov bound with small scale factor (extended abstract), in *28th International Symposium on Theoretical Aspects of Computer Science*, vol. 9 of LIPIcs. Leibniz International Proceedings of Informatics, pp. 201–212 (Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2011)
45. S. Cannon, E. Demaine, M. Demaine, S. Eisenstat, M.J. Patitz, R. Schweller, S.M. Summers, A. Winslow, Two hands are better than one (up to constant factors): self-assembly in the 2HAM vs. aTAM, in *30th International Symposium on Theoretical Aspects of Computer Science*, vol. 20 of LIPIcs. Leibniz International Proceedings of Informatics, pp. 172–184 (Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2013)
46. M. Göös, T. Lempäinen, E. Czeizler, P. Orponen, Search methods for tile sets in patterned DNA self-assembly. *J. Comput. System Sci.* **80**(1), 297–319 (2014)
47. R. Schweller, A. Winslow, T. Wylie, “mplexities for high-temperature two-handed tile self-assembly, in *DNA Computing and Molecular Programming*, vol. 10467 of Lecture Notes in Computer Science, pp. 98–109 (Springer, Cham, 2017)
48. D. Doty, T.A. Rogers, D. Soloveichik, C. Thachuk, D. Woods, Thermodynamic binding networks, in *DNA Computing and Molecular Programming*, vol. 10467 of Lecture Notes in Computer Science, pp. 249–266 (Springer, Cham, 2017)
49. M. Göös, T. Lempäinen, E. Czeizler, P. Orponen, Search methods for tile sets in patterned DNA self-assembly. *J. Comput. Syst. Sci.* **80**(1), 297–319 (2014)
50. R. Veneziano, S. Ratanalert, K. Zhang, F. Zhang, H. Yan, W. Chiu, M. Bathe, et al., Designer nanoscale DNA assemblies programmed from the top down. *Science* (2016)
51. J. Ellis-Monaghan, A. McDowell, I. Moffatt, G. Pangborn, DNA origami and the complexity of Eulerian circuits with turning costs. *Nat. Comput.* **14**(3), 491–503 (2015)
52. N. Jonoska, M. Saito, *Boundary Components of Thickened Graphs, Revised Papers of 7th International Meeting on DNA Based Computers*, vol. 2340, pp. 70–81 (Springer LNCS, 2002)
53. W. Wang, S. Chen, B. An, et al., Complex wireframe DNA nanostructures from simple building blocks. *Nat. Commun.* **10**, 1067 (2019)
54. S.W. Bent, U. Manber, On nonintersecting Eulerian circuits. *Discrete Appl. Math.* **18**(1), 87–94 (1987)
55. N. Jonoska, N.C. Seeman, G. Wu, On existence of reporter strands in DNA-based graph structures. *Theoret. Comput. Sci.* **410**(15), 1448–1460 (2009)
56. M.N. Ellingham, J.A. Ellis-Monaghan, Edge-outer graph embedding and the complexity of the DNA reporter strand problem. *Theoret. Comput. Sci.* **785**, 117–127 (2019)
57. D. Archdeacon, Variations on a theme of kuratowski. *Discrete Math.* **302**, 22–31 (2005)
58. C. Thomassen, On the complexity of finding a minimum cycle cover of a graph. *Siam J. Comput.* **26**(3), 675–677 (1997)

59. P.W.K. Rothmund, Design of DNA origami, in *Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design*, pp. 471–478 (IEEE Computer Society, 2005)
60. J. Garrett, N. Jonoska, H. Kim, M. Saito, Algebraic systems motivated by DNA origami, in *Algebraic Informatics*, ed. by M. Ćirić, M. Droste, J.-É. Pin, pp. 164–176 (Springer International Publishing, 2019)
61. J. Garrett, N. Jonoska, H. Kim, M. Saito, DNA origami words, graphical structures and their rewriting systems. *Nat. Comput.*, pp. 1–15 (2021)
62. I. Dolinka, J. East, The idempotent-generated subsemigroup of the Kauffman monoid. *Glasgow Math. J.* **59**(3), 673–683 (2017)
63. W. Liu, *Design and Assembly of DNA Nano-Objects and 2D DNA Origami Arrays*. Ph.D. thesis (New York University, 2012)
64. N.C. Seeman, The design of single-stranded nucleic acid knots. *Mol. Eng.* **2**(3), 297–307 (1992)
65. J.L. Ramírez Alfonsín, Knots and links in spatial graphs: a survey. *Discrete Math.* **302**(1–3), 225–242 (2005)
66. N. Robertson, P. Seymour, R. Thomas, Sachs’ linkless embedding conjecture. *J. Combin. Theory Ser. B* **64**(2), 185–227 (1995)
67. J. Conway, C. Gordon, Knots and links in spatial graphs. *J. Graph Theor.* **7**, 445–453 (1983)
68. N. Goldberg, T. Mattman, R. Naimi, Many, many more intrinsically knotted graphs. *Algebr. Geom. Topol.* **14**, 1801–1823 (2014)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

