Summer@ICERM 2023 Project Descriptions

Leyda Almodóvar, Amanda Harsy, Cory Johnson, Jessica Sorrells

Focus of the program: The proposed Summer@ICERM 2023 research topics include projects in a variety of subtopics in graph theory and linear algebra stemming from the flexible-tile model of DNA self-assembly. Projects include computational experimentation and coding to efficiently generate and test hypotheses concerning theoretical optimal constructions of DNA nanostructures.

1 Graph Theoretical Model of DNA Self-Assembly

DNA self-assembly is a term to describe the process of a collection of branched DNA molecules bonding together, without external direction, to form a desired nanostructure. This process is integral to the field of DNA nanotechnology [22]. Synthetic DNA molecules have been designed that self assemble into a variety of nanostructures, including numerous polyhedra [4, 12, 13, 24, 28], arbitrary graphs [16, 21, 25], and the first macroscopic self-assembled 3D DNA crystals [29]. Molecular scaffoldings made of DNA have wide-ranging potential, such as biomolecular computing and in drug-delivery methods (see [1, 5, 7, 8, 11, 17, 18, 19, 20, 23, 26, 27]).

The introduction of a graph theoretical formalization for exploring the combinatorial properties of self-assembly of DNA molecules was first introduced by Jonoska et al. in [15]. A *branched junction molecule* whose flexible k-arms are strands of DNA is represented in the abstract as a *tile*, which becomes a vertex of degree k in a graph (see Figure 1). We represent the complementary cohesive-ends or *bond-edges* of these tiles using letter labels. For example, bond-edge types a and $\hat{a}$ represent complementary sequences of bases (see Figures 2 and 4). Thus we can combinatorially represent a k-armed branched-junction molecule with bond-edge types $a_1, ..., a_k$ using a tile $t = \{a_1, ..., a_k\}$. We call a collection of tiles (each of which can theoretically be used "infinitely many" times) a *pot* (see Figure 3).

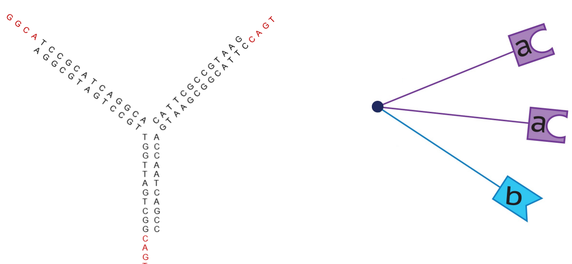

Figure 1: *3*-armed branched junction molecule (left) with example tile representation (right)

Figure 2: Representing the complementary cohesive end types

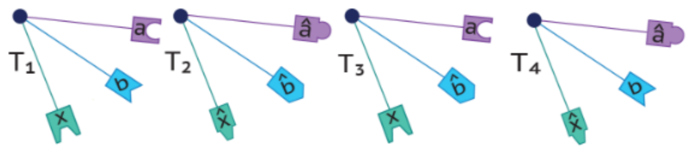

Figure 3: A pot of 4 different tile types, $P = \{T_1 = \{a, b, x\}, T_2 = \{\hat{a}, \hat{b}, \hat{x}\}, T_3 = \{a, \hat{b}, x\}, T_4 = \{\hat{a}, b, \hat{x}\}\}$.

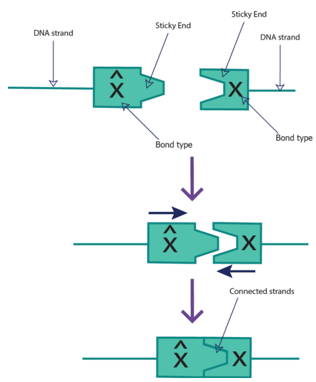

Figure 4: A visualization of the formation of bond-edges.

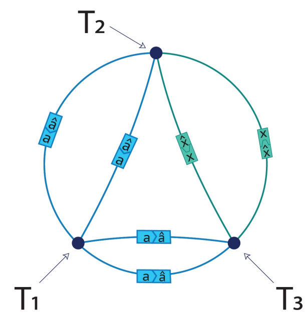

Figure 5: Example of a graphical representation of a complete complex consructed from pot $P = \{T_1 = \{a^4\}, T_2 = \{\hat{a}^2, x, \hat{x}\}, T_3 = \{\hat{a}^2, x, \hat{x}\}\}$.

The central focus of research with the flexible-tile model of DNA self-assembly is efficient construction of certain target complexes (see Figures 5 and 6). This generally involves finding accurate bounds for the number of tile types and bond-edge types in pots constructing selected graphs. Given a target graph, G, one can seek to determine the minimum numbers of tile and bond types needed to construct the graph; we denote these minimum numbers by $T_i(G)$ and $B_i(G)$, respectively, where i refers to one of the scenarios below.

Scenario 1: A graph with a fewer number of vertices than the target graph may be constructed from a pot of tiles.
Scenario 2: A graph with the same number of vertices, but not isomorphic to the target graph, may be constructed from a given pot of tiles. However, no graph with fewer vertices may be constructed.
Scenario 3: No graph with fewer vertices nor non-isomorphic graphs with the same number of vertices as the target graph may be constructed from a given pot of tiles.

The projects proposed here revolve around the exploration of the graph theoretical and combinatorial properties of DNA self-assembly, as well as development of computational tools to aid in answering fundamental questions that arise.

2 Project Descriptions

2.1 Optimal Pots for k-Regular Graphs

In 2014 Ellis-Monaghan et. al. [6] showed that for any k-regular graph G, $T_1(G) = 1$ or 2 if k is even or odd, respectively. Some examples of k-regular graphs have been explored in Scenarios 2 and 3, such as cycle graphs, Platonic Solid graphs, and the Rook's graph [2, 6, 9]. However, no overarching theory regarding general optimization strategies for k-regular graphs in Scenarios 2 and 3 currently exists. We propose the following items for investigation.

1. Find explicit pots of tiles for certain families of k-regular graphs such as crown graphs, cage graphs, and n-circulant graphs.

2. For fixed values of k, search for patterns common to pots of tiles for k-regular graphs and establish conjectures regarding bounds for $B_2(G)$, $T_2(G)$, $B_3(G)$, and $T_3(G)$.

3. Collaborate with those working on projects described in Sections 2.3 and 2.4 to computationally verify pots and further progress.

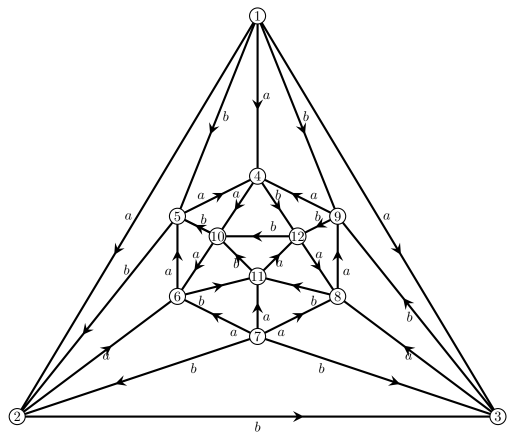

Figure 6: Scenario 2 labeling of Icosahedron graph, a 3-regular graph

2.2 Optimal Pots for Graph Families Exhibiting Multi-Dimensional Growth

Families of graphs that exhibit growth in two or more distinct ways or "dimensions" have proven especially challenging when determining optimal pots for Scenarios 2 and 3. For example, in [10] it is shown that for the Lollipop and Tadpole graph families (see Figure 7) the values for Scenario 2 have nuanced dependence on both the order of the complete or cycle subgraph and the extending path subgraph. We propose the following items for investigation.

1. Discover explicit pots of tiles for certain graph families with multi-dimensional growth, such as lattice graphs, Mongolian tent graphs, and stacked book graphs.

2. Search for patterns common to pots of tiles for graph families exhibiting multi-dimensional growth. Determine how pots must be modified to accommodate "growth" in one isolated aspect.

3. Collaborate with those working on projects described in Sections 2.3 and 2.4 to computationally verify pots and further progress.

2.3 Expected Output of Pots

Much of the existing work within the flexible-tile model of DNA self-assembly has focused on finding an optimal pot that will construct a given graph [2, 6]. Whenever exploring this question in Scenarios 2 and 3, one must consider whether other graphs of lesser or equal orders can be constructed from proposed pots. However, little is known about the general problem of what types of graphs can be constructed from a given pot. In [2], it was shown that determining the set of graphs realized by a given pot is an NP-hard problem.

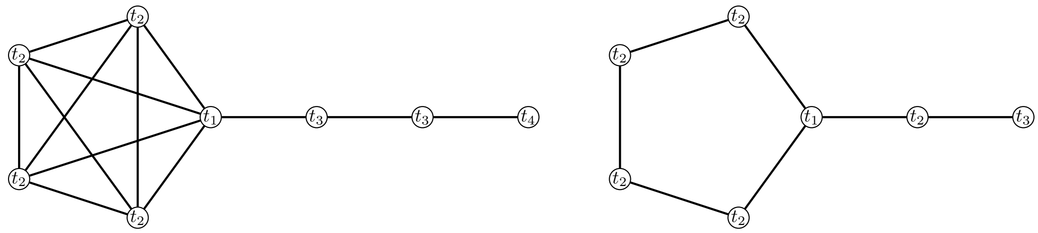

Figure 7: Lollipop graph (left) and Tadpole graph (right)

For this reason, it is desirable to be able to classify graphs in the output of a pot in special settings. In [14], two algorithms are presented which construct a graph for any pot with exactly one bond-edge type and at least three tile types. We propose the following items for investigation.

1. Given a pot with two or more bond-edge types, find an algorithm that will construct at least one type of graph realized by the pot of tiles.

2. Generate examples of graphs that can be constructed from the pots of known graphs such as complete bipartite graphs, platonic solids, cycle graphs, etc.

3. Find relationships between the construction matrices of different graphs within the same family, further clarifying which orders of graphs can be constructed by the corresponding pots.

2.4 Algorithmic Generation of Pots

Finding an optimal pot to realize a target graph requires sorting through a generic list of possible tile types. This list grows quickly as the number of bond-edge types and/or vertex degree increases. For example, when considering two bond-edge types, there are 10 different tiles of degree two. To this end, a useful program would be one that takes in the degree sequence of a graph and a desired number of bond-edge types and outputs all possible tile types given those constraints. To build upon this foundation, the program can be enhanced to omit some types of tiles that are always impermissible in Scenarios 2 or 3. For example, the tile type $\{a, \hat{a}\}$ is generally forbidden in Scenarios 2 and 3, as a graph of order 1 with a loop edge or a graph of order 2 with a multiple-edge can be constructed from this tile (see Figure 8).

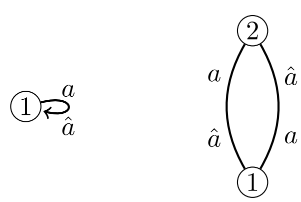

Figure 8: Examples of order 1 and 2 graphs constructed from tile type $\{a, \hat{a}\}$

In addition, some sets of two or more tiles constitute immediate violations in these scenarios; consider tiles $\{a, b\}$ and $\{\hat{a}, \hat{b}\}$, which can construct a graph of order 2. We propose the following coding initiatives to support these explorations.

1. Write a program that generates all possible tile types that could be used in the construction of a graph when given a degree sequence and maximum number of bond-edge types.

2. Modify such a program to avoid pots with subsets consisting of tile types that are "automatic" failures in Scenarios 2 and 3.

2.5 Discovering Relationships Between $B_i(G)$ and $T_i(G)$ for $i = 2, 3$

Theorem 2 of [6] states that $T_2(G) \geq B_2(G) + 1$, which serves as an incredibly useful bound when working in Scenario 2. Indeed, for many basic graph families, such as cycle graphs and complete graphs, the values of $B_i(G)$ and $T_i(G)$ follow the pattern $T_i(G) = B_i(G) + 1$ for $i = 2, 3$. However, graphs exist for which $T_2(G) - B_2(G)$ is arbitrarily large [6] and for which the values for Scenarios 2 and 3 differ by up to a factor of four [3]. Further complexity is illustrated in [2], where an example is given of a graph G of order 6 (see Figure 9) for which no pot can simultaneously achieve the values for $B_3(G)$ and $T_3(G)$.

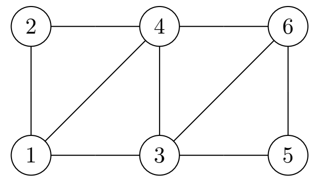

Figure 9: 2×3 triangle lattice graph

Meanwhile, for every graph that has been investigated in the literature, $B_2(G)$ and $T_2(G)$ *have* been achieved by the same pot. This naturally provokes questions about whether there can be any persistent relationship between $B_i(G)$ and $T_i(G)$ for $i = 2, 3$, and about types of graphs for which no pot achieving both exists. We propose the following items for investigation:

1. Determine whether there exist a pot that will achieve $B_2(G)$ and $T_2(G)$ simultaneously for a given graph G. Prove that such a pot always exists or find a counterexample.
2. Generate examples of graphs of various orders for which no pot achieves both $B_3(G)$ and $T_3(G)$; provide proofs that no such pots exist.
3. Collect and analyze existing data on values of $B_3(G)$ and $T_3(G)$ for various graphs and graph families and search for meaningful possible relationships or convincing arguments.

References

[1] Leonard M Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.

[2] Leyda Almodóvar, Jo Ellis-Monaghan, Amanda Harsy, Cory Johnson, and Jessica Sorrells. Computational complexity and pragmatic solutions for flexible tile-based dna self-assembly. Under review, http://arxiv.org/abs/2108.00035.

[3] Leyda Almodóvar, Jo Ellis-Monaghan, Amanda Harsy, Cory Johnson, and Jessica Sorrells. Optimal tile-based dna self-assembly designs for lattice graphs and platonic solids. 2021. https://scholarworks.lib.csusb.edu/mathematics-publications/1/.

[4] Junghuei Chen and Nadrian C Seeman. Synthesis from dna of a molecule with the connectivity of a cube. *Nature*, 350(6319):631, 1991.

[5] J. Ellis-Monaghan, N. Jonoska, and G. Pangborn. Tile-based dna nanostructures: Mathematical design and problem encoding. In R. Robeva and M. MaCauley, editors, *Algebraic and Computational Biology*. Elsevier, 2019.

[6] Joanna Ellis-Monaghan, Greta Pangborn, Laura Beaudin, David Miller, Nick Bruno, and Akie Hashimoto. Minimal tile and bond-edge types for self-assembling dna graphs. In *Discrete and Topological Models in Molecular Biology*, pages 241–270. Springer, 2014.

[7] M. Ferrari, A. Cook, A. Houlihan, R. Roulaeu, N. Seeman, G. Pangborn, and J. Ellis-Monaghan. Design formalism for dna self-assembly of polyhedral skeletons using rigid tiles. *The Journal of Mathematical Chemistry*, 56(5):1365–1392, 2018.

[8] Andre K Geim and Konstantin S Novoselov. The rise of graphene. In *Nanoscience and Technology: A Collection of Reviews from Nature Journals*, pages 11–19. World Scientific, 2010.

[9] Ernesto Gonzalez. Tile based self-assembly of the rook's graph. *Electronic Theses, Projects, and Dissertations*, 2020. https://scholarworks.lib.csusb.edu/etd/1085.

[10] D. Chloe Griffin and Jessica Sorrells. Tile based modeling of dna self-assembly for two graph families with appended paths. *Involve*. To Appear, https://arxiv.org/abs/2203.07343.

[11] Hongzhou Gu, Jie Chao, Shou-Jun Xiao, and Nadrian C Seeman. A proximity-based programmable dna nanoscale assembly line. *Nature*, 465(7295):202, 2010.

[12] Yu He, Tao Ye, Min Su, Chuan Zhang, Alexander E Ribbe, Wen Jiang, and Chengde Mao. Hierarchical self-assembly of dna into symmetric supramolecular polyhedra. *Nature*, 452(7184):198, 2008.

[13] Ryosuke Iinuma, Yonggang Ke, Ralf Jungmann, Thomas Schlichthaerle, Johannes B Woehrstein, and Peng Yin. Polyhedra self-assembled from dna tripods and characterized with 3d dna-paint. *science*, page 1250944, 2014.

[14] Cory Johnson and Andrew Lavengood-Ryan. Dna complexes of one bond-edge type. In Preparation, https://scholarworks.lib.csusb.edu/etd/1039.

[15] Nataša Jonoska, Gregory L McColm, and Ana Staninska. Spectrum of a pot for dna complexes. In *International Workshop on DNA-Based Computers*, pages 83–94. Springer, 2006.

[16] Nataša Jonoska, Phiset Sa-Ardyen, and Nadrian C Seeman. Computation by self-assembly of dna graphs. *Genetic Programming and Evolvable Machines*, 4(2):123–137, 2003.

[17] Hyunho Kim, Sungwoo Yang, Sameer R Rao, Shankar Narayanan, Eugene A Kapustin, Hiroyasu Furukawa, Ari S Umans, Omar M Yaghi, and Evelyn N Wang. Water harvesting from air with metal-organic frameworks powered by natural sunlight. *Science*, 356(6336):430–434, 2017.

[18] Thom H LaBean and Hanying Li. Constructing novel materials with dna. *Nano Today*, 2(2):26–35, 2007.

[19] Kyle Lund, Anthony J Manzo, Nadine Dabby, Nicole Michelotti, Alexander Johnson-Buck, Jeanette Nangreave, Steven Taylor, Renjun Pei, Milan N Stojanovic, Nils G Walter, et al. Molecular robots guided by prescriptive landscapes. *Nature*, 465(7295):206, 2010.

[20] Tosan Omabegho, Ruojie Sha, and Nadrian C Seeman. A bipedal dna brownian motor with coordinated legs. *Science*, 324(5923):67–71, 2009.

[21] Phiset Sa-Ardyen, Nataša Jonoska, and Nadrian C Seeman. Self-assembling dna graphs. In *International Workshop on DNA-Based Computers*, pages 1–9. Springer, 2002.

[22] Nadrian C Seeman. An overview of structural dna nanotechnology. *Molecular biotechnology*, 37(3):246, 2007.

[23] Nadrian C Seeman. An overview of structural dna nanotechnology. *Molecular biotechnology*, 37(3):246, 2007.

[24] Wei Sun, Etienne Boulais, Yera Hakobyan, Wei Li Wang, Amy Guan, Mark Bathe, and Peng Yin. Casting inorganic structures with dna molds. *Science*, page 1258361, 2014.

[25] Gang Wu, Natasha Jonoska, and Nadrian C Seeman. Construction of a dna nano-object directly demonstrates computation. *Biosystems*, 98(2):80–84, 2009.

[26] Hao Yan, Sung Ha Park, Gleb Finkelstein, John H Reif, and Thomas H LaBean. Dna-templated self-assembly of protein arrays and highly conductive nanowires. *Science*, 301(5641):1882–1884, 2003.

[27] Peng Yin, Hao Yan, Xiaoju G Daniell, Andrew J Turberfield, and John H Reif. A unidirectional dna walker that moves autonomously along a track. *Angewandte Chemie*, 116(37):5014–5019, 2004.

[28] Yuwen Zhang and Nadrian C Seeman. Construction of a dna-truncated octahedron. *Journal of the American Chemical Society*, 116(5):1661–1669, 1994.

[29] Jianping Zheng, Jens J Birktoft, Yi Chen, Tong Wang, Ruojie Sha, Pamela E Constantinou, Stephan L Ginell, Chengde Mao, and Nadrian C Seeman. From molecular to macroscopic via the rational design of a self-assembled 3d dna crystal. *Nature*, 461(7260):74, 2009.