Minimal Tile and Bond-Edge Types for Self-Assembling DNA Graphs of Triangular Lattice Graphs

Leyda Almodóvar
Dept. of Mathematics
Stonehill College
Easton, MA 02357, USA

Sydney Martin
Dept. of Mathematics
Stonehill College
Easton, MA 02357, USA

Samantha Mauro
Dept. of Mathematics
Stonehill College
Easton, MA 02357, USA

Heiko Todt
Dept. of Mathematics
Stonehill College
Easton, MA 02357, USA

Abstract

We utilize the flexible tile model presented in [13] to design self-assembling DNA structures from a graph theory perspective. These tiles represent branched junction molecules whose arms are double strands of DNA. We consider $2\times n$ triangular lattice graphs G_n, where n represents the number of triangles. Given a target graph G_n, we determine the minimum number of tile and bond-edge types needed in order to create G_n as a complete self-assembled complex in three different scenarios. Each scenario corresponds to a distinct level of laboratory constraint. In the first scenario, graphs of a smaller size than G_n are allowed. In the second scenario, non-isomorphic graphs of the same size as G_n are allowed, but not graphs of smaller size. In the third scenario, only graphs isomorphic or larger in size to the target graph are allowed. We provide optimal tile sets for all $2 \times n$ triangular lattice graphs G_n in Scenario 1 and Scenario 3. We also include some small examples in Scenario 2.

CONGRESSUS NUMERANTIUM 232 (2019), pp.241-264

1 Introduction

DNA self-assembly is an emerging and rapidly advancing field, with references [22, 24] providing good overviews. Synthetic DNA molecules have been designed that self assemble into nanostructures in a particular shape, starting with branched DNA molecules [14, 27], nanoscale arrays [28, 29], numerous polyhedra [3, 9, 10, 25, 32], arbitrary graphs [11, 2, 30], a variety of DNA and RNA knots [17, 18, 26], and the first macroscopic self-assembled 3D DNA crystals [33]. These nanostructures have a wide range of applications such as containers for the transport and release of nanocargos, templates for the controlled growth of nano-objects, biomolecular computing, biosensors and in drug-delivery methods [1, 7, 8, 15, 16, 19, 21, 23, 31, 6, 4]. While several self-assembly methods exist [3, 12], we focus on the flexible-tile model introduced in [13] and we expand on the results presented in [5].

In this paper, we focus on the graph-theoretical aspect of designing the building blocks that will construct $2 \times n$ triangular lattice graphs of DNA structures. In the flexible tile model these building blocks are star-shaped molecules whose flexible k-arms are double strands of DNA as shown in Figure 1 [20]. These arms have cohesive ends that can bond to any other cohesive end with a complementary Watson-Crick base. We call the k-armed molecule a "tile" and represent it as a vertex of degree k in a graph. We represent the complementary cohesive-ends or *bond-edges* of these tiles using letter labels, where complementary sequences of bases are represented by hatted and unhatted letters. For example, given the bond end type a, its complementary sequences of bases is represented by $\hat{a}$. We say a collection of tiles, called a *pot* realizes a graph, G, if the collection of tiles constructs the same structure as G. The tile type is the multiset of letters corresponding to the cohesive-end types for the tile. We require no unmatched cohesive ends in the resulting DNA complex.

The central point of our research is to find the minimum number of bond-edge and tile types needed in order to realize a target graph G_n, where G_n represents a triangular lattice graph and n represents the number of triangles. An example is shown in Figure 2.

We consider these numbers under three different scenarios corresponding to three laboratory constraints:

- **Scenario 1.** The incidental construction of a graph smaller than G_n is allowed.
- **Scenario 2.** The incidental construction of a graph smaller than G_n is not allowed but a non-isomorphic graph with the same size as G_n is allowed.

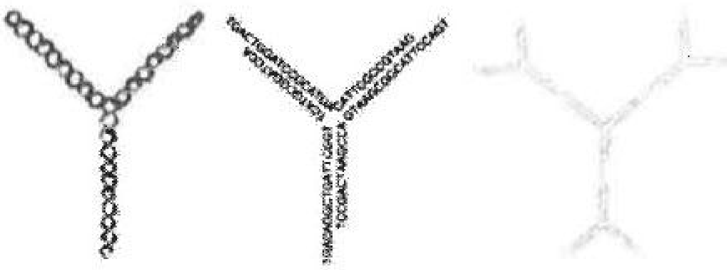

Figure 1: k-armed branched junction molecules

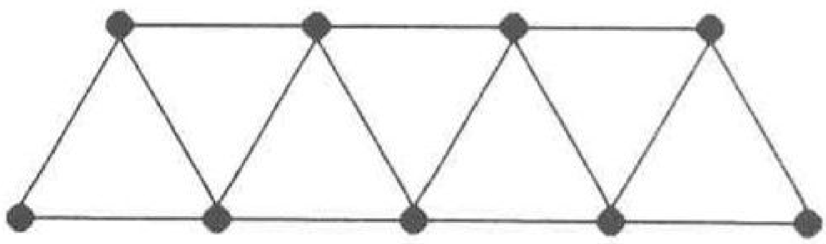

Figure 2: The triangular lattice graph G_7

- **Scenario 3.** Any non-isomorphic graph incidentally constructed must be larger than G_n.

An example how these scenarios differ can be found in Example 1.1.

Additionally, we let $T_i(G_n)$ for $i = 1, 2, 3$ be the minimum number of tiles required to construct a complex in each of the scenarios above. Similarly, $B_i(G_n)$ denotes the minimum number of bond-edge types needed for each scenario. Currently, known results include optimal solutions for cycles, trees, complete graphs and bipartite graphs for all scenarios [5]. We built upon these results and found optimal solutions for $2 \times n$ triangular lattice graphs in Scenarios 1 and 3. We also include some results for small graphs in Scenario 2.

Example 1.1. *Suppose we want to construct the triangular lattice graph G_2 with two triangles. The pot $P = \{\{a, \hat{a}\}, \{a^2, \hat{a}\}, \{a, \hat{a}^2\}\}$ realizes G_2 as can be seen in Figure 3. However, this pot does not satisfy Scenarios 2 or 3 as the tile $\{a, \hat{a}\}$ can also realize the graph with a single a vertex and a loop by connecting the ends a and $\hat{a}$.*

The pot $P = \{\{a.b\}, \{\hat{a}^2\}, \{a^2, \hat{a}\}, \{a, \hat{a}, \hat{b}\}\}$ realizes G_2 as can be seen in Figure 4. It is not possible to realize a graph with less than four vertices with this pot. However, it is possible to construct the graph in Figure 5, which is not isomorphic to G_2. Therefore, this pot satisfies Scenario 2, but not Scenario 3.

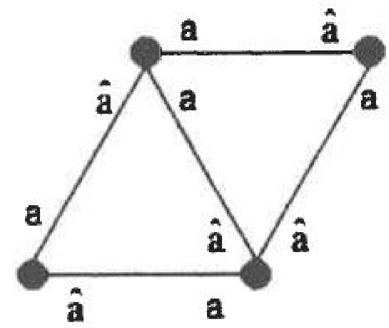

Figure 3: A construction of G_2 that satisfies Scenario 1

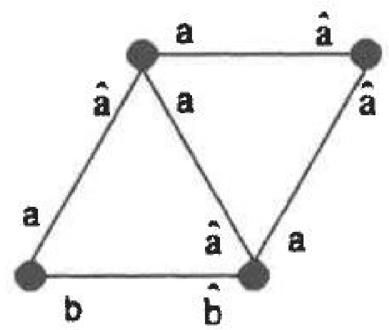

Figure 4: A construction of G_2 that satisfies Scenario 2

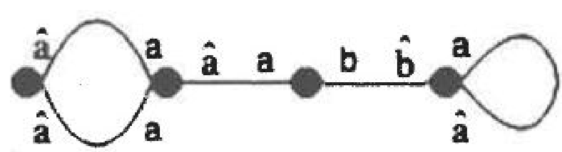

Figure 5: A graph realized by the same pot as G_2 in Scenario 2

Finally, the pot $P = \{\{a, b\}, \{\hat{a}^2, b\}, \{\hat{b}^3\}\}$ *realizes* G_2 *as can be seen in Figure 6. It can be checked that any graph that can be realized by this pot has at least four vertices and if it has four vertices it is isomorphic to* G_2*. Therefore, this pot satsifies Scenario 3.*

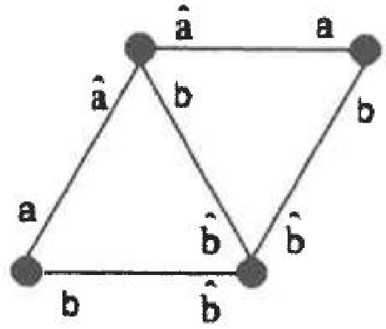

Figure 6: A construction of G_2 that satisfies Scenario 3

2 Scenario 1

It is clear that $B_1(G) = 1$ for any graph G, since one can just use the same bond-edge type for every edge of the graph. Therefore, we only have to determine $T_1(G_n)$. We prove the following theorem for triangular lattice graphs.

Theorem 2.1. *$T_1(G_1) = 1$, $T_1(G_2) = 3$, $T_1(G_3) = 3$, $T_1(G_4) = 4$, $T_1(G_5) = 3$. If $n \geq 6$, then $T_1(G_n) = 4$.*

Proof. We have $T_1(G_1) = 1$, since the pot $P = \{\{a, \hat{a}\}\}$ realizes G_1.

The triangular lattice graph with two triangles, G_2, has two vertices of degree two and two vertices of degree three. Therefore, $T_1(G_2) \geq 2$. However, $T_1(G_2) \neq 2$, since it is impossible to use the same tile twice for both of the degree two and degree three vertices. A degree two tile has a difference between the hatted and unhatted bond-edge types of either -2, 0 or 2. A degree three tile has a difference between the hatted and unhatted bond-edge types of either -3, -1, 1 or 3. If each tile is used twice, the sum of the differences cannot be equal to zero. However, in order to be a complete complex the difference between the hatted and unhatted bond-edge types in a graph has to be zero.

The pot $P = \{\{a^2\}, \{a, \hat{a}^2\}, \{\hat{a}^3\}\}$ realizes G_2 as can be seen in Figure 7. Since $T_1(G_2) > 2$ and we find a pot with three tiles, we conclude that $T_1(G_2) = 3$.

The triangular lattice graph with three triangles, G_3, has two vertices of degree two, two vertices of degree three and one vertex of degree four. Therefore, $T_1(G_3) \geq 3$. The pot $P = \{\{a^2\}, \{a, \hat{a}^2\}, \{a, \hat{a}^3\}\}$ realizes G_3 and has three tiles as can be seen in Figure 8. Therefore, $T_1(G_3) = 3$.

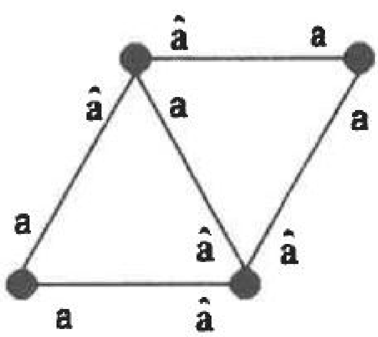

Figure 7: A construction of G_2 that satisfies Scenario 1

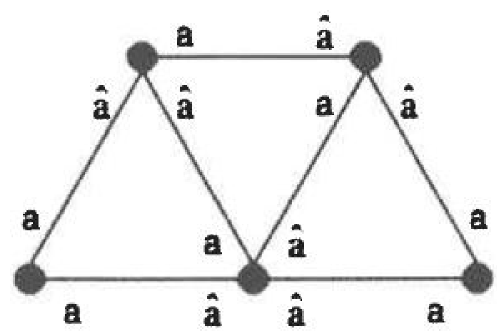

Figure 8: A construction of G_3 that satisfies Scenario 1

The triangular lattice graph with four triangles, G_4, has two vertices of degree two, two vertices of degree three and two vertices of degree four. Therefore, $T_1(G_4) \geq 3$. With a little effort, one can show that at least one of the tiles has to be repeated, so $T_1(G_4) > 3$. The pot $P = \{\{a^2\}, \{a^2, \hat{a}\}, \{a, \hat{a}^3\}, \{\hat{a}^4\}\}$ realizes G_4 and has four tiles as can be seen in Figure 9. Therefore, $T_1(G_4) = 4$.

The triangular lattice graph with five triangles, G_5, has two vertices of degree two, two vertices of degree three and three vertices of degree four. Therefore, $T_1(G_5) \geq 3$. The pot $P = \{\{a^2\}, \{a^2, \hat{a}\}, \{a, \hat{a}^3\}\}$ realizes G_5 and has three tiles as can be seen in Figure 10. Therefore, $T_1(G_5) = 3$.

If $n \geq 6$, the triangular lattice graph G_n has two vertices of degree two, two vertices of degree three and $n-2$ vertices of degree four. Therefore, $T_1(G_n) \geq 3$. Suppose $T_1(G_n) = 3$. Then the pot consists of exactly one tile of degree two, one tile of degree three and one tile of degree four. Any tile of degree two has a difference of 0 or ± 2 between the hatted and unhatted bond-edge types. Any tile of degree three has a difference of ± 1 or ± 3 between the hatted and unhatted bond-edge types. Any tile of degree four

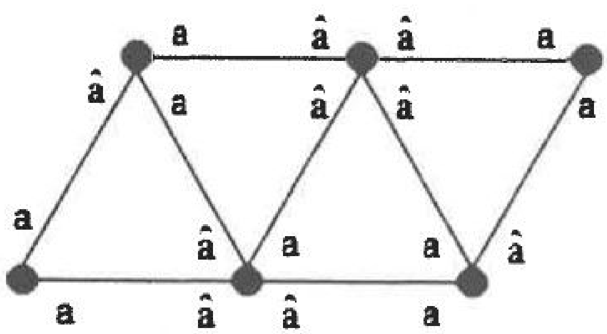

Figure 9: A construction of G_4 that satisfies Scenario 1

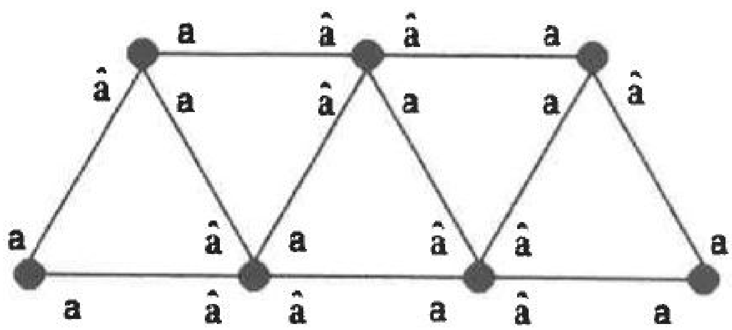

Figure 10: A construction of G_5 that satisfies Scenario 1

has a difference of 0, ± 2 or ± 4 between the hatted and unhatted bond-edge types. If $T_1(G_n) = 3$, then the degree two tile is used twice, the degree three tile is used three times and the degree four tile is used $n-2$ times. The differences between hatted and unhatted bond-edge types in a complete complex has to be zero. We can therefore express this as an equation and check whether this equation has a solution. The equation is

$$2x + 2y + (n-2)z = 0$$

where $x = 0, \pm 2$, $y = \pm 1, \pm 3$ and $z = 0, \pm 2, \pm 4$.

First note that $z \neq \pm 4$, since then $|(n-2)z| \geq 16$, but $|2x + 2y|$ is at most 10.

If $z = \pm 2$, there is a solution only if $n = 7$, namely $x = 2$, $y = 3$ and $z = -2$. However, this would mean that the degree two and degree three tiles both have only hatted or unhatted bond-edge types. Since in G_n the degree two and degree three vertices are connected by an edge this is a contradiction.

We can therefore deduce that $z = 0$. But then the equation reduces to $2x + 2y = 0$, which can be simplified to $x + y = 0$. Since $x = 0, \pm 2$ and $y = \pm 1, \pm 3$, we have reached a contradiction. Therefore, if $n \geq 6$, $T_1(G_n) > 3$.

The pot $P = \{\{a^2\}, \{a, \hat{a}\}, \{a, \hat{a}^2\}, \{\hat{a}^4\}\}$ realizes G_n and has four tiles. Figures 11 and 12 show the construction and this can easily be generalized to G_n with more more triangles. Therefore, if $n \geq 6$, $T_1(G_n) = 4$.

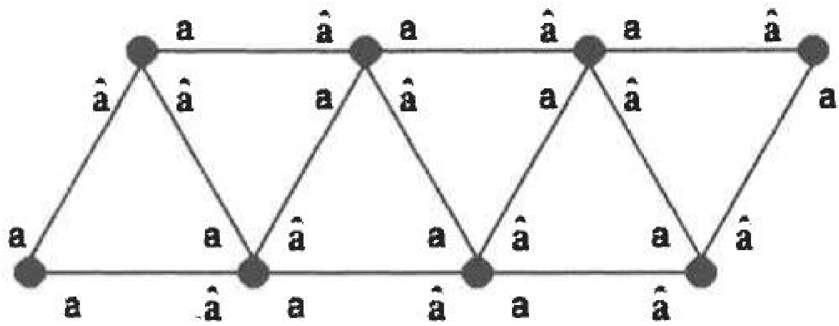

Figure 11: A construction of G_6 that satisfies Scenario 1

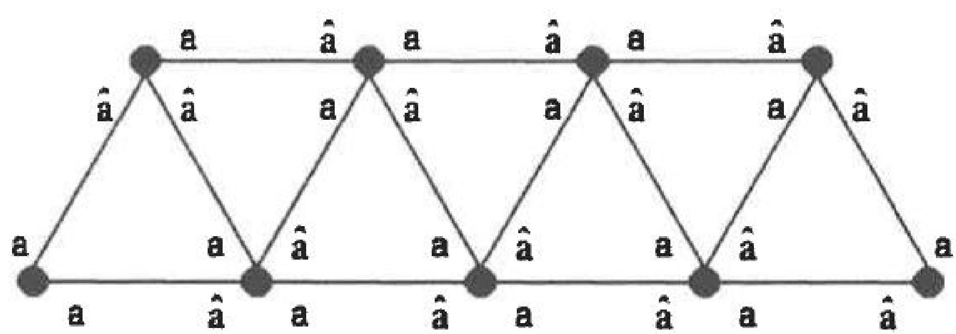

Figure 12: A construction of G_7 that satisfies Scenario 1

□

3 Scenario 3

The following lemma from [5] is useful to determine which bond-edge types can be repeated.

Lemma 3.1. *If P is a pot such that $\{G\} = C_{\min}(P)$, and two nonadjacent edges $\{u, v\}$ and $\{s, t\}$ of $G = \{V, E\}$ use the same bond-edge type, then G is isomorphic to $G' = \{V, E'\}$, where $E' = E - \{\{u, v\}, \{s, t\}\} \cup \{\{u, t\}, \{s, v\}\}$.*

Figure 13 shows how edges with the same bond-edge type are replaced according to Lemma 3.1. Replacing edges in that manner has to yield an isomorphic graph, otherwise the pot does not satisfy scenario 3.

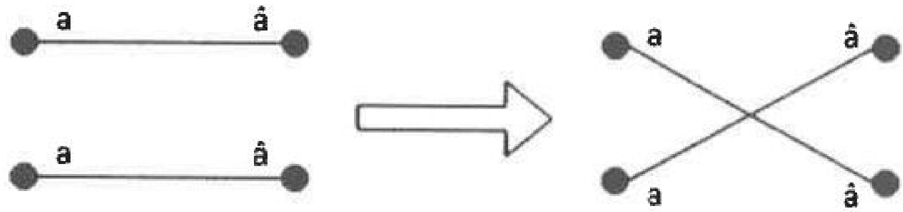

Figure 13: Switching edges according to 3.1

Using this lemma, we prove the following lemma for triangular lattice graphs.

Lemma 3.2. *In a triangular lattice graph with at least five triangles, a bond-edge type can only be repeated on the same tile or as shown in Figure 14.*

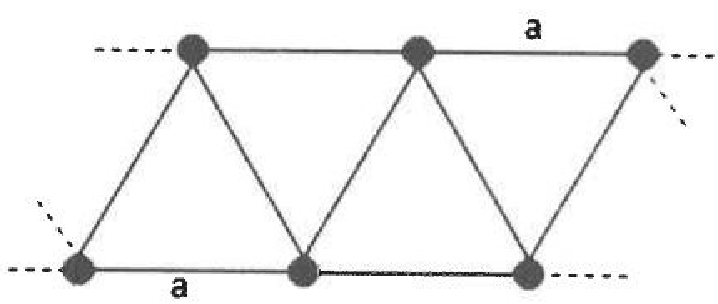

Figure 14: Repeated bond-edge type on different tiles

Note that the dotted lines mean that there may be more triangles to the left or right, but there do not have to be.

Proof. We first note that a bond-edge type can be repeated on the same tile, as long as the tile includes exclusively the hatted bond-edge type or exclusively the unhatted bond-edge type. In that case no other nonisomoprhic graph can be created from the pot.

Thus, we only have to consider repeating a bond-edge type on edges that are not adjacent. By Lemma 3.1, if a bond-edge type is repeated, removing the two edges and reconnecting the vertices with different edges as stated in the lemma has to result in an isomorphic graph.

We introduce some terminology to discuss the different edges in a triangular lattice graph. Note that a triangular lattice graph allows exactly one Hamiltonian circuit. We call the edges that form this Hamiltonian circuit the *outside edges*, except for the two edges connecting a degree two vertex with a degree three vertex, which are called the *side edges*. The edges that are not part of the Hamiltonian circuit are called the *inside edges*. Note that this nomenclature requires us to consider triangular lattice graphs with at least five triangles to distinguish between outside edges and side edges. We will also later see that a smaller number of triangles allows for certain special cases. Figure 15 shows the different types of edges of a triangular lattice graph.

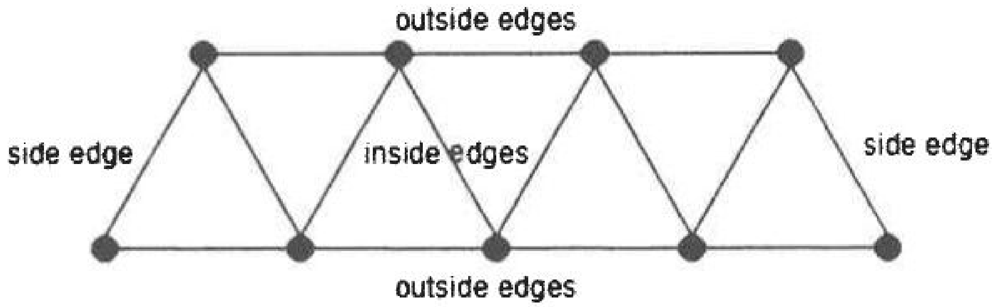

Figure 15: Different edge types

Now suppose an inside edge is repeated elsewhere in the graph, not on the same tile. As this inside edge is then removed according to Lemma 3.1 it will leave a cycle of length four in the graph where none of the opposite vertices are connected. This type of cycle does not exist in a triangular lattice graph, thus the resulting graph is not isomorphic. Therefore, no inside edge can be repeated elsewhere in the graph except on the same tile.

As a consequence, we only have to consider repeated bond-edge types where both edges are outside edges or side edges. We first consider consider two outside edges on the same side as in Figure 16.

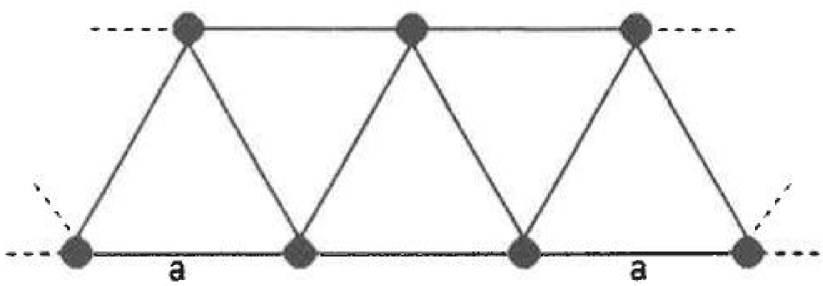

Figure 16: Two outside edges on the same side

Applying Lemma 3.1 creates the graph in Figure 17. Note that the other possible graph created by Lemma 3.1 would have a double-edge and is definitely not isomorphic to a triangular lattice graph.

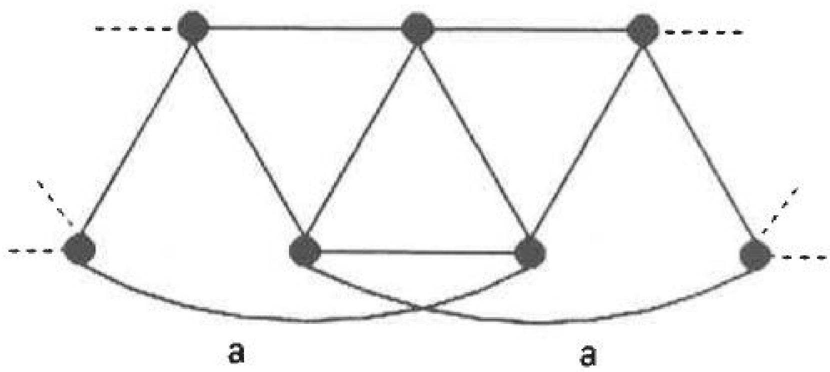

Figure 17: Two outside edges on the same side

We notice that this graph contains a cycle of length six with only two of these vertices also connected with another edge. This does not exist in a triangular lattice graph, so the resulting graph is not isomorphic. If the two outside edges on the same side are even further apart, it will create a cycle of length six where no vertices are connected with another edge, so this also is not isomorphic to a triangular lattice graph.

We can therefore restrict ourselves to outside edges that are on opposite sides.

Let us first consider the following two outside edges on opposite sides in Figure 18 that are close to each other.

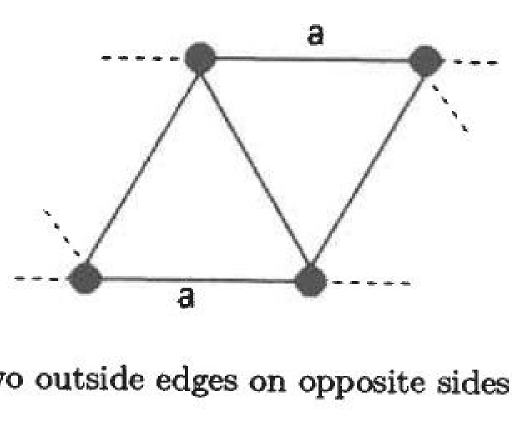

Figure 18: Two outside edges on opposite sides that are close

We see that the only two ways to apply Lemma 3.1 yield the following two graphs in Figure 19.

Both of these graphs have double-edges, so the result is not isomorphic.

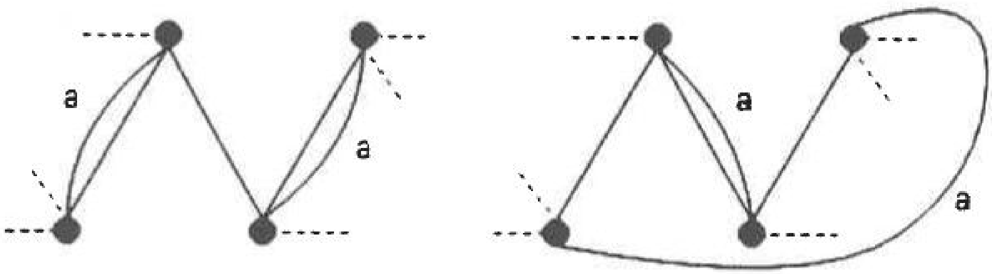

Figure 19: Applying Lemma 3.1 to two close opposite edges

Let us now consider two outside edges that are further apart from each other as in Figure 20.

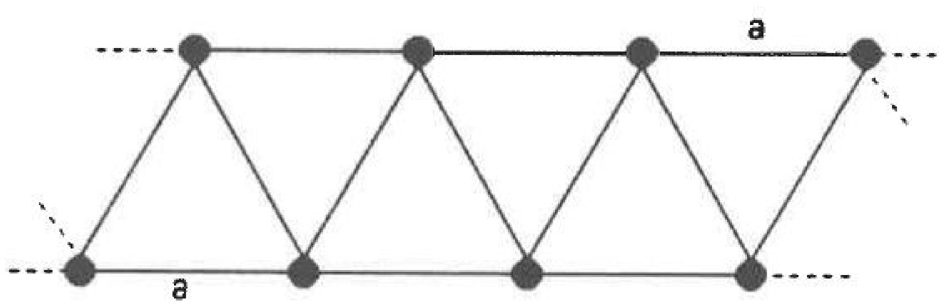

Figure 20: Two outside edges on opposite sides that are far apart

The two possible graphs from Lemma 3.1 are the ones shown in Figures 21 and 22. Neither one of these graphs is isomorphic to a triangular lattice graph, since both decrease the distance between the vertices of degree two.

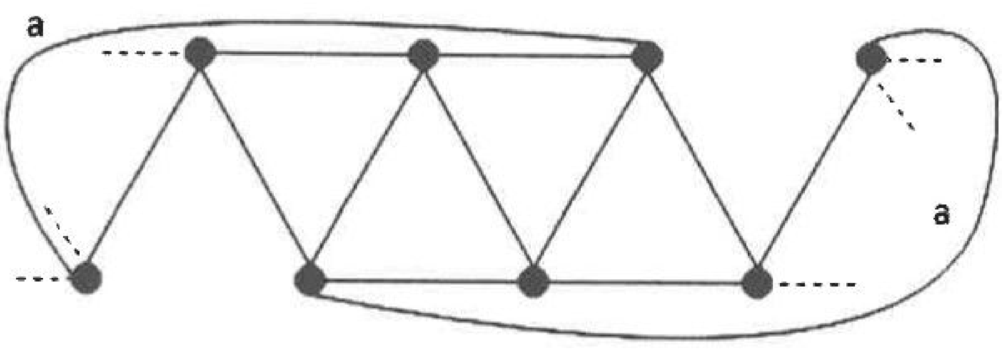

Figure 21: First possibility from Lemma 3.1 applied to two outside edges that are far apart

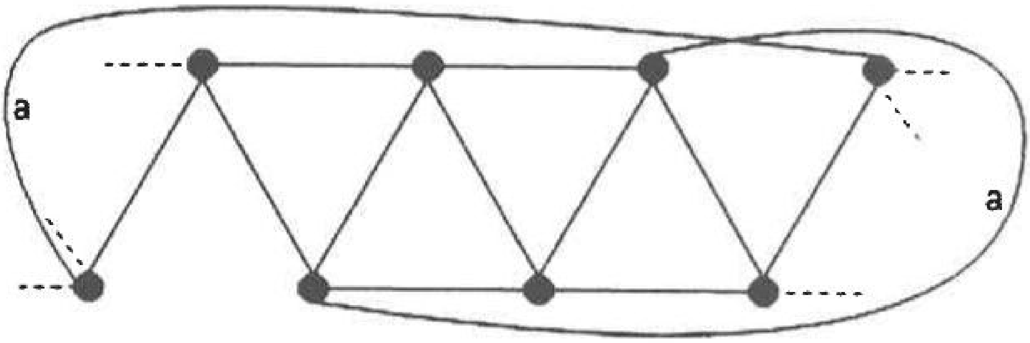

Figure 22: Second possibility from Lemma 3.1 applied to two outside edges that are far apart

A similar argument works for two outside edges that are even further apart than the previous case.

If we consider the cases of one side edge having the same bond-edge type as an outside edge or the two side edges having the same bond-edge type, similar arguments apply to show that we obtain a graph that is not isomorphic to a triangular lattice graph.

We now turn our attention to the most interesting case, namely having repeated bond-edge types as in Figure 14.

We remove and reattach edges as in Lemma 3.1 and obtain the graph in Figure 23.

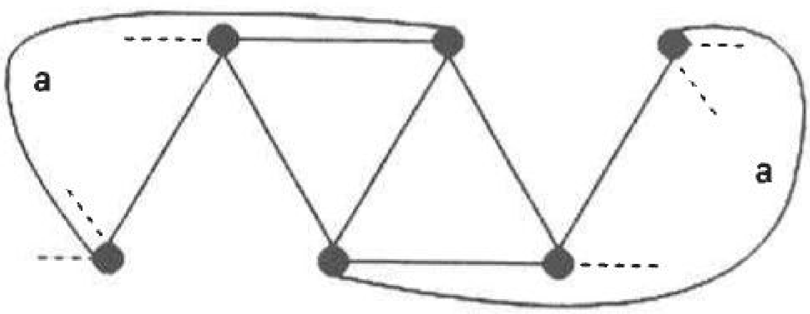

Figure 23: Second possibility from Lemma 3.1 applied to two outside edges that are far apart

This graph is indeed isomorphic to a triangular lattice graph. It is therefore possible to repeat a bond-edge type for this particular configuration, in addition to repeating bond-edge types on the same tile.

□

An immediate consequence of Lemma 3.2 is the following theorem.

Theorem 3.3. *If $n \geq 5$, then $T_3(G_n) = n + 2$.*

Proof. By Lemma 3.2, we see that no tile can be repeated in a triangular lattice graph with at least five triangles. Note that even if a bond-edge type is repeated as in Figure 14, the tiles cannot be repeated because that would imply that a repeated bond-edge type would appear on an inside edge. Since a triangular lattice graph with n triangles has $n + 2$ vertices, the theorem follows. □

We now just state the results for T_3 for triangular lattice graphs with less than five triangles and give pots that realize them. It is then straightforward to verify that no smaller pot satisfies scenario 3.

Theorem 3.4. $T_3(G_1) = 3$, $T_3(G_2) = 3$, $T_3(G_3) = 4$ *and* $T_3(G_4) = 4$.

G_1 is a complete graph so the result follows from [5]. We still give the pot and graph for it for completeness.

A minimal pot for G_1 is $P = \{\{a, b\}, \{\hat{a}^2\}, \{a, \hat{b}\}\}$ and the construction is shown in Figure 24.

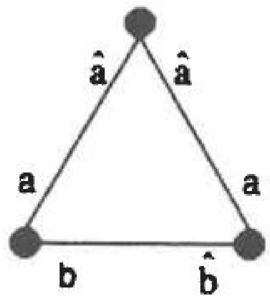

Figure 24: Construction of G_1

A minimal pot for G_2 is $P = \{\{a, b\}, \{\hat{a}^2, b\}, \{\hat{b}^3\}\}$ and the construction is shown in Figure 25.

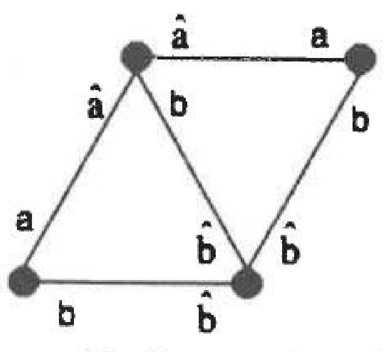

Figure 25: Construction of G_2

A minimal pot for G_3 is $P = \{\{a, b\}, \{\hat{a}, b, c\}, \{\hat{a}, b, \hat{c}\}, \{\hat{b}^4\}\}$ and the construction is shown in Figure 26.

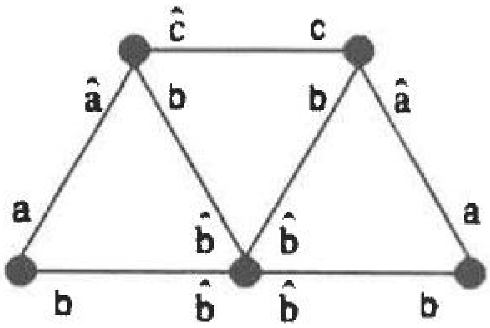

Figure 26: Construction of G_3

A minimal pot for G_3 is $P = \{\{a, b\}, \{\hat{a}, \hat{c}, \hat{d}\}, \{\hat{b}, c^3\}, \{\hat{b}, \hat{c}, d^2\}\}$ and the construction is shown in Figure 27.

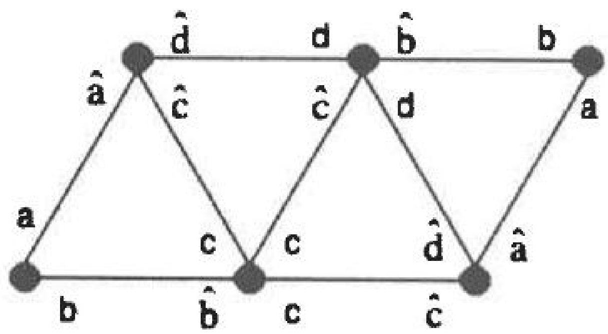

Figure 27: Construction of G_4

We see that in G_4 the special case of Lemma 3.2 is used to create a pot with less tiles than one would expect. This pot also takes advantage of obtaining an isomorphic graph if the first triangle and the last triangle are moved to the top and bottom respectively when applying Lemma 3.1. This allows the side edges to be repeated.

This explains that there is a sharp increase in the minimum tiles needed from G_4 to G_5, namely $T_3(G_4) = 4$, but $T_3(G_5) = 7$.

Lemma 3.2 can also be used to determine B_3 for triangular lattice graphs with at least five triangles. The result is stated in the next theorem.

Theorem 3.5. *If $n \geq 5$, then $B_3(G_n) = \lceil \frac{3n+3}{4} \rceil$ if n is odd and $B_3(G_n) = \lceil \frac{3n+2}{4} \rceil$ if n is even.*

Proof. From Lemma 3.2 we know that bond-edge types can only be repeated on the same tile or in the configuration as in Figure 14. For determining the minimum number of bond types for at least five triangles, this special configuration can be ignored. Note it will not allow less bond types used when compared to reusing a bond-edge on the same tile, where it possibly can be reused even more often.

We therefore focus our attention to reusing the same bond-edge type on the same tile as often as possible. All of the inside and side edges have to be labeled, so in order to do this most efficiently, we use the same label for the two inside/side edges from the same vertex. We then may as well label any attached outside edges with the same bond-type. Figures 28 and 29 show this process.

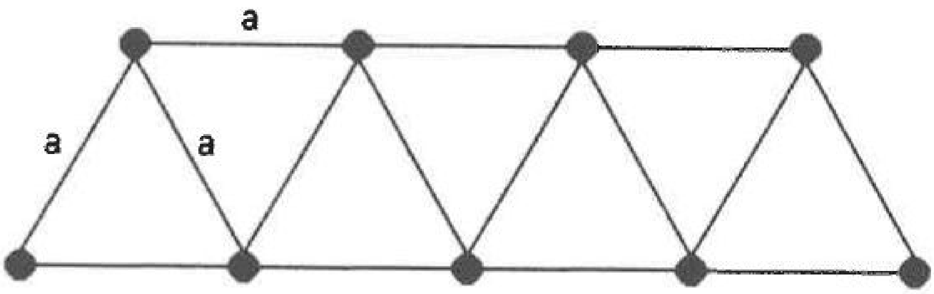

Figure 28: Labeling of a side edge and an inside edge with the same bond-edge type

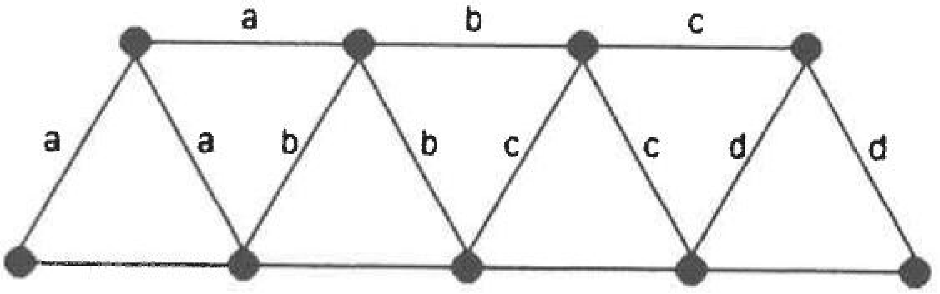

Figure 29: Labeling of the side edges and all inside edges

This process allows us minimize the number of bond-edge types used for the inside edges, side edges and upper outside edges. For example, we need four bond-edge types to label these edges for G_7. In general, we need $\frac{n+1}{2}$ bond-edge types to label inside, side and upper outside edges for G_n.

We now just have label the lower outside edges and we do this in pairs of two edges attached to the same vertex to minimize the number of bond-edge types. This is shown in Figure 30 for G_7.

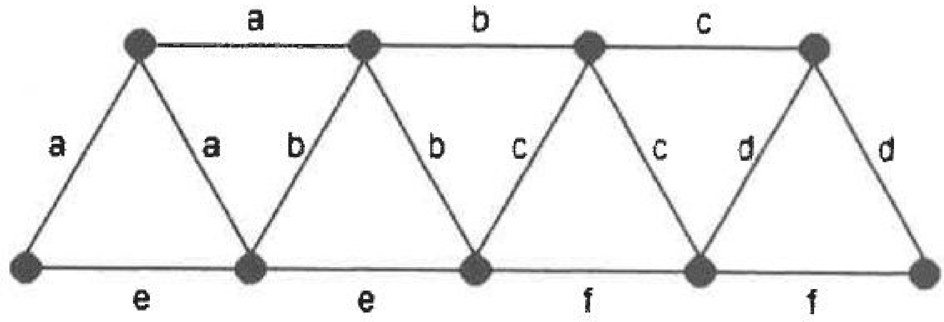

Figure 30: Complete labeling of G_7

We see that we need an additional two bond-edge types for G_7. In general, we have $\frac{n+1}{2}$ lower outside edges, so we need an additional $\lceil\frac{n+1}{4}\rceil$ bond-edge types.

If n is odd, we therefore need a total of $\frac{n+1}{2}+\lceil\frac{n+1}{4}\rceil = \lceil\frac{3n+3}{4}\rceil$ bond-edge types.

We use a similar idea if n is even. We first use the minimum number of bond-edge types for the inside and upper outside edges. Note that since n is even, the right side edge cannot be labeled this way. This is shown for G_8 in Figure 31.

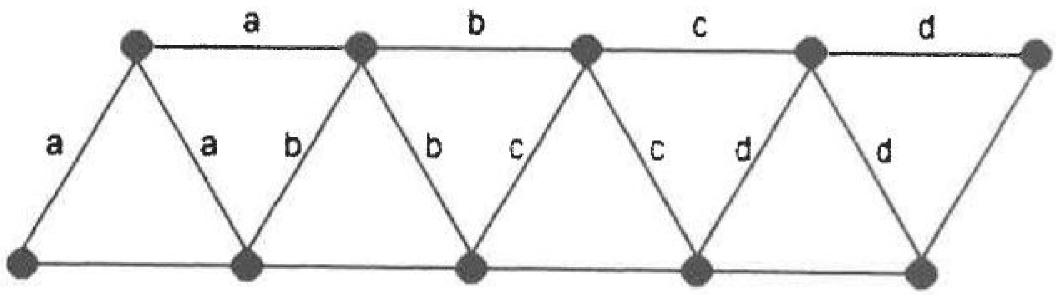

Figure 31: Labeling of the inside edges and upper outside edges

There are four bond-edge types needed for this for G_8, and in general it would be $\frac{n}{2}$ for G_n.

We then label the remaining lower outside edges and the right side edge in pairs of two as show in Figure 32.

For G_8 we need three bond-edge types to label the remaining five edges. In general, there are $\frac{n}{2}+1$ edges remaining, so for G_n we need an additional $\lceil\frac{n+2}{4}\rceil$ bond-edge types. If n is even we then need a total of $\frac{n}{2}+\lceil\frac{n+2}{4}\rceil = \lceil\frac{3n+2}{4}\rceil$ bond-edge types.

□

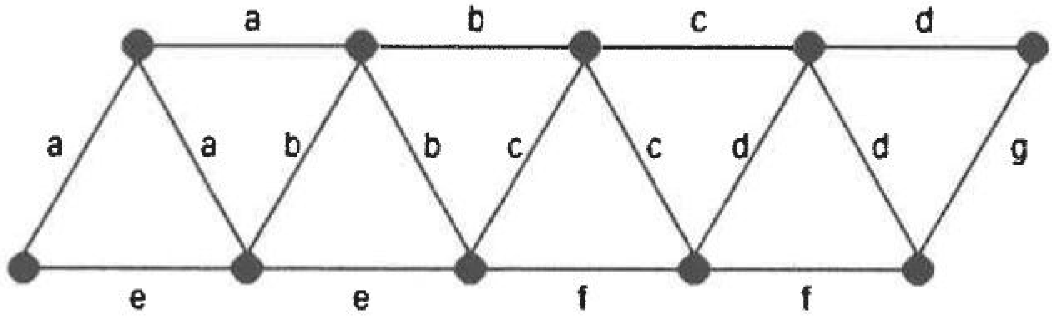

Figure 32: Complete labeling of G_8

We complete Scenario 3 by stating the results for B_3 for triangular lattice graphs with less than five triangles. It is straightforward to verify these.

Theorem 3.6. *$B_3(G_1) = 2$, $B_3(G_2) = 2$, $B_3(G_3) = 3$ and $B_3(G_4) = 3$.*

The same pots as in Figures 24, 25 and 26 can be used for $B_3(G_1)$, $B_3(G_2)$ and $B_3(G_3)$. For G_4 the pot that minimizes the number of tiles does not minimize the number of bond-edge types, so we have to use a different pot. A pot that minimizes the number of bond-edge types for G_4 is $P = \{\{a,b\},\{a,c\},\{\hat{a},b,c\},\{\hat{b}^4\},\{b,\hat{c}^3\}\}$ and the construction is shown in Figure 33.

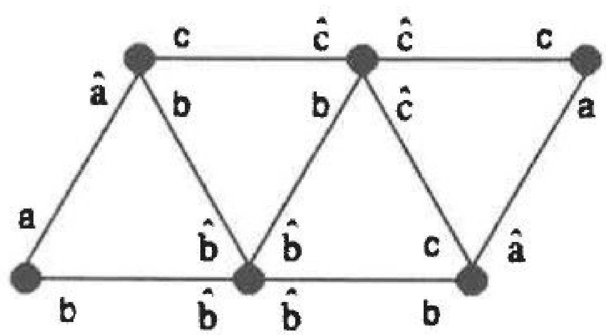

Figure 33: Construction of G_4

4 Scenario 2

We found the minimum number of tile and bond-edge types for several small graphs. The construction matrix presented in [5] was employed in order to show that a smaller graph than the target graph cannot be created from a given pot. Since we know the minimum number of tiles needed for each graph in Scenario 1 and Scenario 3, the bound $T_1(G) \leq T_2(G) \leq T_3(G)$ presented in [5] granted $T_2(G_n)$ for some graphs.

4.1 G_2

Since we found $T_1(G_2) = T_3(G_2) = 3$, we must have $T_2(G_2) = 3$ as well. The pot $P = \{\{a, b\}, \{a, \hat{b}\}, \{\hat{a}, b, \hat{b}\}\}$ realizes G_2 and no smaller graph can be constructed from this pot. It is then straightforward to verify that $B_2(G_2) = 2$.

4.2 G_3

Since there are vertices of degree 2, 3, and 4 respectively, we know $T_2(G_3) \geq 3$. The pot $P = \{\{\hat{a}, b\}\{a, b, \hat{b}\}, \{a, \hat{a}, \hat{b}^2\}\}$ realizes G_3.

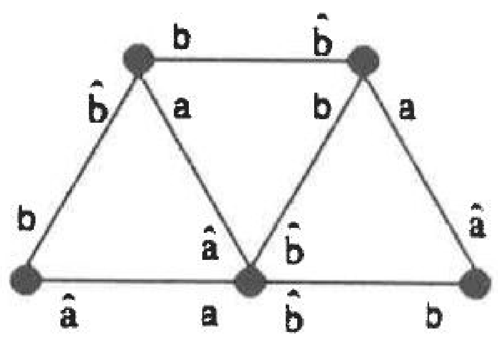

Figure 34: Construction of G_3

Since the construction matrix for this pot yields a unique solution, we know that there is no smaller graph that can made from this pot. Thus, $T_2(G_3) = 3$. We also found $B_2(G_3) = 2$.

4.3 G_4

Since $T_1(G_4) = T_3(G_4) = 4$, we know $T_2(G_4) = 4$.

The pot for the graph pictured is $P = \{\{\hat{a}^2\}, \{a^2, b\}, \{a, \hat{a}, \hat{b}, c\}, \{a, \hat{a}, \hat{b}, \hat{c}\}\}$.

Since the construction matrix for this pot yields a unique solution, we know that there is no smaller graph that can made from this pot. We also found that $B_2(G_4) = 2$ with $P = \{\ \{\hat{a}, b\}, \{\hat{a}, \hat{b}\}, \{a^3\}, \{\hat{a}^2, b^2\}, \{\hat{a}^2, \hat{b}^2\}\}$ realizing G_4.

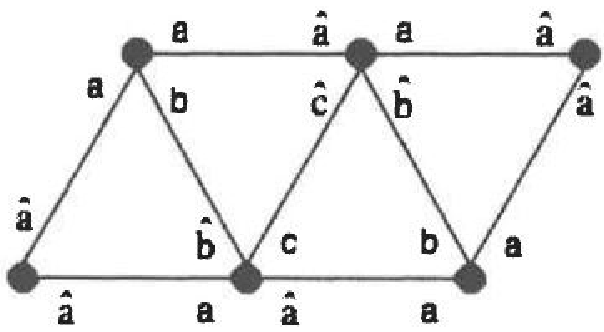

Figure 35: Construction of G_4

4.4 G_5

Even though $T_2(G_4) = 4$, we found that $T_2(G_5) = 3$. The pot $P = \{\{a, b\}, \{\hat{a}, b^2\}, \{\hat{b}^2, c, \hat{c}\}\}$ realizes G_5 and it is straightforward to show that no smaller graph can be created from this pot.

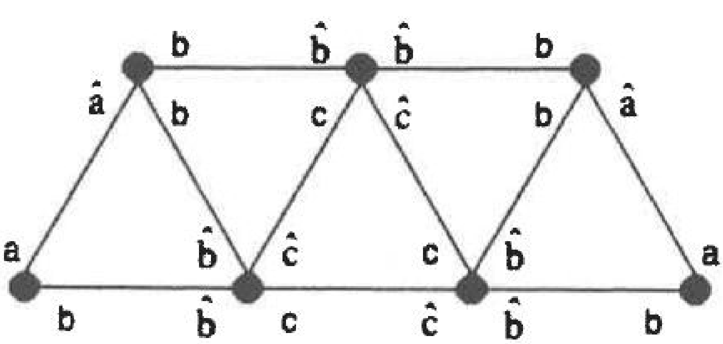

Figure 36: Construction of G_5 minimizing $T_3(G_5)$

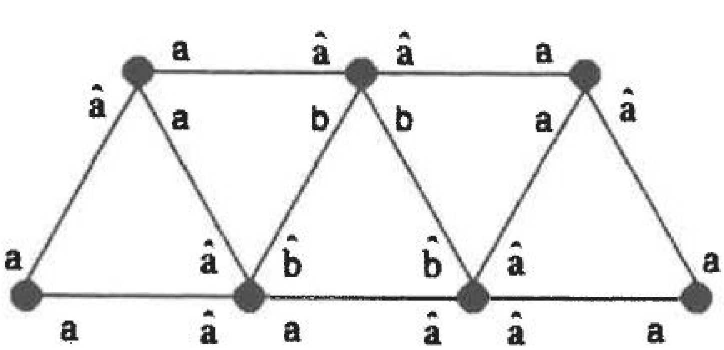

Figure 37: Construction of G_5 minimizing $B_3(G_5)$

5 Conclusion

Although we found optimal pots for $2 \times n$ triangular lattice graphs in Scenario 1 and Scenario 3, we still need to determine an efficient way to design pots in Scenario 2. Particularly, we need to explore whether the behavior for the minimum number of tiles from four triangles to five triangles is a special case, or whether we need to consider even and odd number of triangles separately. Also, we provided some examples in Scenario 2 and 3 were it is not possible to find a pot that simultaneously achieved both the minimum number of bond-edge types and the minimum number of tile types. We need to explore conditions under which a pot will simultaneously achieve both minimums.

6 Acknowledgments

We would like to thank the Stonehill Undergraduate Research Experience (SURE) program for support and funding to make this paper possible.

References

[1] Leonard M Adleman. "Molecular computation of solutions to combinatorial problems". In: *Nature* 369 (1994), p. 40.

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

[3] Junghuei Chen and Nadrian C Seeman. "Synthesis from DNA of a molecule with the connectivity of a cube". In: *Nature* 350.6319 (1991), pp. 631–633.

[4] J. Ellis-Monaghan, N. Jonoska, and G. Pangborn. "Tile-based DNA Nanostructures: Mathematical Design and Problem Encoding". In: *Algebraic and Computational Biology.* Ed. by R. Robeva and M. MaCauley. Elsevier.

[5] J Ellis-Monaghan et al. "Minimal tile and bond-edge types for self-assembling DNA graphs". In: *Discrete and Topological Models in Molecular Biology.* Springer, 2014, pp. 241–270.

[6] M. Ferrari et al. "Design formalism for DNA self-assembly of polyhedral skeletons using rigid tiles". In: *The Journal of Mathematical Chemistry* 56.5 (2018), pp. 1365–1392.