Maximum Packings in Tripartite Graphs

by

You Rao

B.Sc. Hons, The University of British Columbia, 2017

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The College of Graduate Studies

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

January 2020

The following individuals certify that they have read, and recommend to the College of Graduate Studies for acceptance, a thesis/dissertation entitled:

MAXIMUM PACKINGS IN TRIPARTITE GRAPHS

submitted by YOU RAO in partial fulfilment of the requirements of the degree of Master of Science

Dr. Wayne Broughton, Irving K. Barber School of Arts and Sciences
Supervisor

Dr. Donovan Hare, Irving K. Barber School of Arts and Sciences
Supervisory Committee Member

Dr. Javad Tavakoli, Irving K. Barber School of Arts and Sciences
Supervisory Committee Member

Dr. Holger Andreas, Irving K. Barber School of Arts and Sciences
University Examiner

Abstract

This thesis is motivated by an attempt to prove a conjecture in design theory due to Hiralal Agrawal, by interpreting it in graph theory as a consequence of a possible extension of Hall's Marriage Theorem to tripartite graphs. Previous work on Agrawal's conjecture has been done from design theory perspective. We define *edge* $\triangle$*-*r*-regular tripartite graphs*, inspired by Agrawal's design theory construction, and consider the conjecture that if one side of a tripartite graph is a minimum transversal of the triangles in the graph, then there exists a packing of triangles in the graph that saturates that side. Although there are counterexamples to this general statement, it has been shown to hold for certain special kinds of tripartite graphs, and we conjecture that it holds for edge $\triangle$-r-regular tripartite graphs when $r \geq 3$. We use techniques in an array representation of the graph to find a maximum packing in this type of graph, but cannot find an effective method to prove that a complete packing exists in general.

We also study edge $\triangle$-2-regular tripartite graphs and show that such a graph satisfies the statement of the conjecture if and only if its triangle graph is bipartite, and that this is also equivalent to the orientability of the triangulated surface defined by the triangles of the graph.

Lay Summary

This thesis is motivated by a construction in the statistical design of experiments which was suggested by Hiralal Agrawal in 1966. He could not prove that his method would work in general, and this is still an open question. We interpret his question in terms of certain kinds of graphs, as finding the largest possible number of triangles in each of these graphs that do not share any edge. We prove some related results and study techniques for constructing the largest possible set of triangles for some of these graphs.

Table of Contents

List of Tables

List of Figures

Acknowledgement

First and foremost, I really appreciate all the help from my supervisor Dr. Wayne Broughton throughout my undergrad and graduate student life. Without his precious advice in both math and life, I would not have the courage to accomplish my Master's degree.

I would like to thank Dr. Donovan Hare and Dr. Javad Tavakoli for being on my committee, Dr. Holger Andreas to be my University Examiner. I would also like to thank Dr. Rebecca Tyson for her great help and encouragement to me to choose to be a math major student. I really appreciate Dr. Qiduan Yang, Dr. Shawn Wang and Dr. Heinz Bauschke for their generous kindness of sharing knowledge with me.

Thanks to UBC Okanagan campus for financial support throughout my graduate studies.

I would also wish to thank Dr. Blair Spearman, who initially accepted me as his student, although this was tragically cut very short. I wish I could have met him earlier. Through him, I got to know his great students: Dr. Paul Lee, Dr. Chad Davis, Dr. Lindsey Reinholz, Stephen Brown and Jeewon Yoo. Thank you all for the help and advice.

I thank all my friends for any kind of help in my life when I am a student and teaching assistant in UBCO, and especially for all the support from my friend Jingshi Guan. I could not get through my hard times without your company.

Last but not the least, I really appreciate all the support from my beloved parents. I am blessed to be your child and your unconditional love is the strongest shelter in my life.

Dedication

To my parents, my aunt and uncle, my friends.
To Dr. Blair Spearman.

Chapter 1

Introduction

In this chapter, we will introduce some basic definitions in graph theory that will be used later in this thesis. The materials in this chapter are mainly from [Wes] and [BM$^+$76]. We will provide some examples of some definitions.

1.1 Basic Concepts of General Graphs

Definition 1.1.1. A *graph* is a mathematical structure consisting of a collection of points called *vertices* (singular: *vertex*) and a collection of unordered pairs of points called *edges*. The points in a pair are called the *endpoints* of the edge and they are joined by the edge. The edge is incident with its endpoints. In this thesis, we will only consider undirected graphs.

For a graph G with n vertices and m edges we denote its vertex set by $V(G) := \{v_1, v_2, \cdots, v_n\}$ and its edge set by $E(G) := \{e_1, e_2, \cdots, e_m\}$.

Definition 1.1.2. Two vertices are *adjacent* if they are connected by an edge. Two adjacent vertices can be connected by more than one edge, which is called multiple edges. If u and v are adjacent, then u is a *neighbour* of v and vice versa. The *neighbourhood* of a vertex is the set of its neighbours

Definition 1.1.3. A *loop* is an edge whose endpoints are the same vertex (considered to be adjacent to itself).

Definition 1.1.4. A graph G is *simple* if it does not contain any multiple edges and loops.

Example 1.1.5. The two graphs drawn below are examples of a simple and a non-simple graph.

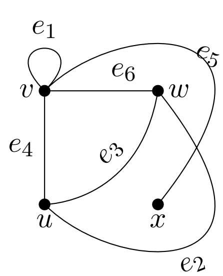
Simple Graph Non-simple Graph

Figure 1.1: Example of simple and non-simple graph

Definition 1.1.6. A graph is *connected* if for any two disjoint non-empty subsets X, Y of the vertex set such that $V = X \bigcup Y$, there exists at least one edge with one endpoint in X and the other endpoint in Y.

Definition 1.1.7. The *adjacency matrix* of a finite graph $G = (V, E)$ (where $|V| = n$ and $|E| = m$) is a square $n \times n$ matrix with each row and column labeled by a vertex. The entry in row u and column v is the number of edges having u and v as endpoints. Note that the adjacency matrix of an undirected graph is symmetric.

Example 1.1.8. Shown below is a finite undirected graph G and its adjacency matrix.



G

	u	v	w	x
u	0	1	2	0
v	1	2	1	1
w	2	1	0	0
x	0	1	0	0

Adjacency matrix of G

Figure 1.2: Example graph and its adjacency matrix

Definition 1.1.9. Let G be a simple graph. A *subgraph* H of G is a graph such that all of the vertices and edges in H are in G; that is, $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A subgraph H is *induced* if every edge in $E(G)$ whose endpoints are in $V(H)$ is also in $E(H)$.

Example 1.1.10. H_1 and H_2 below are two subgraphs of G.

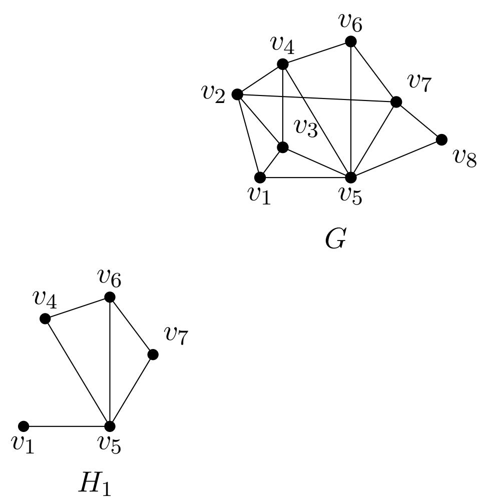


Figure 1.3: Example of induced subgraph and non-induced subgraph

We can see that: $V(G) = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ and $V(H_1) = V(H_2) = \{v_1, v_4, v_5, v_6, v_7\} \subset V(G)$. However, H_2 does not contain all the edges of G whose endpoints are in $V(H_2)$, such as the v_4v_6 and v_6v_7, while H_1 includes all edges of G joining vertices in $V(H_1)$. So H_1 is an induced subgraph of G and H_2 is not.

Definition 1.1.11. Let G be a simple graph. G is *complete* if every pair of distinct vertices in G is adjacent to each other.

Definition 1.1.12. Let G be a simple graph and let $S \subset V(G)$. If every vertex in S is not adjacent to any other vertex in S, we say S is an *independent set*. If $V(G)$ can be partitioned into two disjoint independent sets X, Y (called *parts*), (so $V(G) = X \cup Y$ and $X \cap Y = \varnothing$), then G is a *bipartite* graph denoted by $G[X, Y]$. If $V(G)$ can be partitioned into three disjoint independent sets, then G is a *tripartite* graph. In general, for any simple graph G, if $V(G)$ can be partitioned into k disjoint independent sets, then G is *k-partite*.

Definition 1.1.13. Let $G[X, Y, Z]$ be a tripartite graph with vertex parts X, Y, Z. From the definition of tripartite graph, the edge set can be partitioned into three sets: $E_{XY} := \{xy \in E(G) | x \in X, y \in Y\}$, $E_{XZ} := \{xz \in E(G) | x \in X, z \in Z\}$, $E_{YZ} := \{yz \in E(G) | y \in Y, z \in Z\}$. Each edge set is a *side* of the tripartite graph.

Definition 1.1.14. A graph is a *complete k-partite graph* if it is a k-partite graph and each vertex is adjacent to every other vertex in the other part.

Example 1.1.15. The usual way to denote a complete graph with n vertices is K_n, and for a complete bipartite graph with parts of size m and l we write $K_{m,l}$. The notation of complete k-partite graph is K with subscripts indicating the sizes of the disjoint parts. Below are examples of complete graphs.

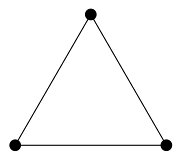


Figure 1.4: K_3

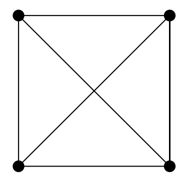


Figure 1.5: K_4

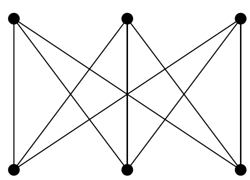


Figure 1.6: $K_{3,3}$

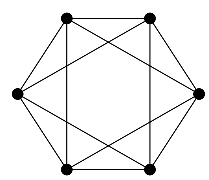


Figure 1.7: $K_{2,2,2}$

Definition 1.1.16. The *degree* of a vertex v is the number of edges which include it as an endpoint, denoted $d(v)$. Every loop is counted twice. For example, every vertex in K_4 has degree 3. If a vertex has degree 0, then it is an *isolated vertex*.

Fact 1.1.17. *For any graph $G = (V, E)$,*

$$\sum_{v \in V} d(v) = 2|E|.$$

Proof. Every edge has two endpoints (possibly same vertex), so every edge is counted twice in the summation of the degrees of all vertices. □

Definition 1.1.18. A graph is *regular* if every vertex has the same degree. A regular graph in which every vertex has degree k is called a k-regular graph. For example, K_4 is a 3-regular graph.

Definition 1.1.19. A graph is *edge regular* if it is a regular graph with every adjacent pair of vertices having exactly the same number of common neighbours. For example, $K_{2,2,2}$ is edge regular.

Definition 1.1.20. Let G be a simple graph. The *complement graph* of G, denoted as $\overline{G}$, is a graph which has the same vertex set $V(G)$, but any two adjacent vertices in G are not adjacent in $\overline{G}$, and those nonadjacent vertices in G are adjacent in $\overline{G}$.

Example 1.1.21. The two graphs below are an example of a graph and its complement.

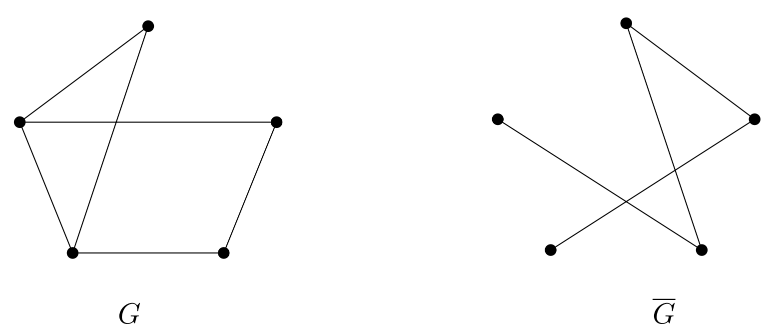


Figure 1.8: Example of complement graph

Note that for any complete graph G, the graph $\overline{G}$ contains no edge.

Definition 1.1.22. Let $G = (V, E)$. A *vertex cover* is a subset of V such that every element in E has at least one endpoint in this subset. If a vertex cover contains the least number of vertices compared to all other vertex covers in G, then it is a *minimum vertex cover*.

Example 1.1.23. On the left below is an example of a vertex cover (in blue) of a graph and on the right is an example of a minimum vertex cover (in red).

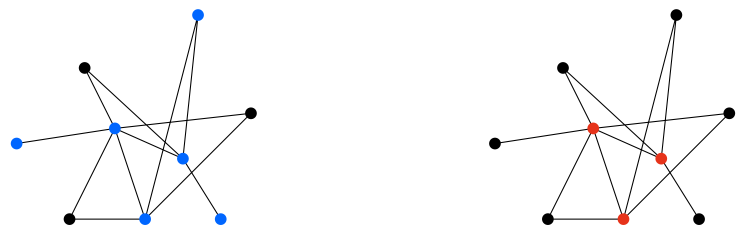


Figure 1.9: Example of a vertex cover and a minimum vertex cover

Definition 1.1.24. Let $G = (V, E)$. An *edge cover* is a subset of E such that every element in V is an endpoint of an edge in the subset. Note that if G contains at least one isolated vertex, then G does not have an edge cover. A *minimum edge cover* of G is an edge cover which has the smallest possible number of elements among all edge covers.

Example 1.1.25. The two diagrams below are examples of an edge cover (in blue) and a minimum edge cover (in red) in a graph.

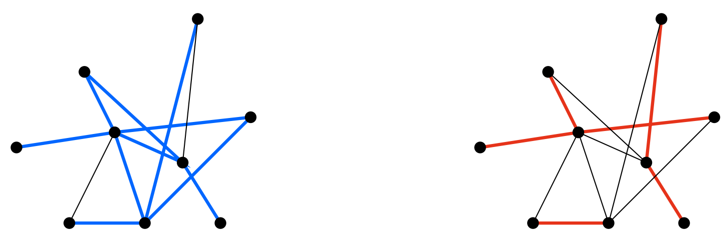


Figure 1.10: Example of an edge cover (blue) and a minimum edge cover (red)

Note that minimum vertex covers and minimum edge covers are not unique in general.

Definition 1.1.26. Let G be a simple graph. A *clique* of G is a subset of vertices in which every vertex in the set is adjacent to every other vertex. In other words, a clique is the vertex set of a complete subgraph and a complement of a clique is an independent set.

Definition 1.1.27. The *clique number* of a simple graph G, denoted by $\omega(G)$, is the maximum number of vertices in a clique in G.

Definition 1.1.28. The *clique covering number* of a simple graph G, denoted by $\theta(G)$, is the minimum cardinality of a set of cliques whose union includes all vertices of G.

Definition 1.1.29. The *chromatic number* of a simple graph G, denoted by $\chi(G)$, is the minimum number of colours required to colour the vertices of G so that no two adjacent vertices receive the same colour. Note that $\chi(G)$ is the smallest number k such that G is k-partite.

Fact 1.1.30. *For any simple graph G, $\chi(G) \geq \omega(G)$.*

Proof. The maximum clique has to be assigned different colours so by definition $\chi(G) \geq \omega(G)$. □

Example 1.1.31. A simple graph G and its complement $\overline{G}$ are given below.

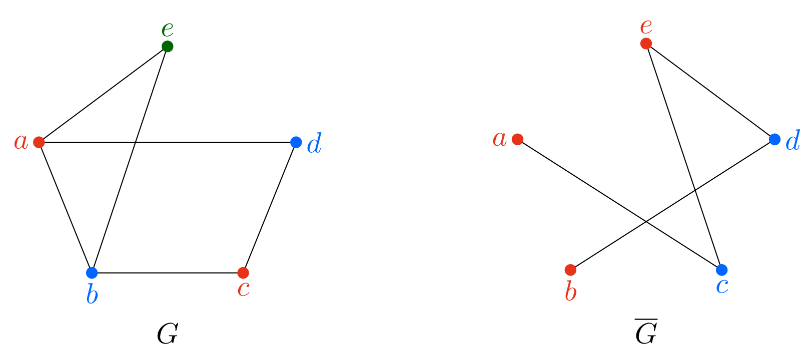


Figure 1.11: Example for θ, ω, χ

We can see that for G, the clique with the maximum number of vertices is $\{a, b, e\}$, so $\omega(G) = 3$. Since $\omega(G) = 3$, we need at least three different colors to cover all the vertices in G and as we can see from the graph, in fact $\chi(G) = 3$. If we take the maximum clique and another clique $\{c, d\}$, then we have included all vertices in G. Thus $\theta(G) = 2$. Looking at $\overline{G}$, we can see $\omega(\overline{G}) = 2$, $\chi(\overline{G}) = 2$, $\theta(\overline{G}) = 3$.

Fact 1.1.32. *For any simple graph G, $\chi(G) = \theta(\overline{G})$.*

Proof. It follows from the definitions. All vertices in a clique of $\overline{G}$ are non-adjacent in G, therefore, a clique in $\overline{G}$ can be assigned same colour in G. Hence the minimum number of cliques that include all vertices in $\overline{G}$ is equal to the minimum number of colours required to colour all vertices of G. □

Definition 1.1.33. A graph G is *perfect* if the chromatic number of every induced subgraph equals the clique number of this subgraph. Equivalently, G is perfect if and only if for all induced subgraphs $H \subseteq G$:

$$\chi(H) = \omega(H)$$

Definition 1.1.34. A *walk* W is a sequence of vertices and edges "$v_0, e_1, v_1, e_2, v_2, \cdots, v_{n-1}, e_n, v_n$", where $v_{i-1}v_i = e_i$ for all $1 \le i \le n$. The *length* of a walk is the number of the edges contained in the walk. A *path* is a walk in which every vertex appears no more than once.

Definition 1.1.35. A *cycle* is a path but with the exception that the starting and ending point are the same vertex. Note that a loop is a cycle of length 1, and a cycle of length 3 is a *triangle*. Briefly, we use *even cycle* for a cycle with even length; and *odd cycle* for a cycle with odd length.

Example 1.1.36. A common notation for a cycle of length n is C_n. Below is a cycle of length 5 (C_5) and its complement.

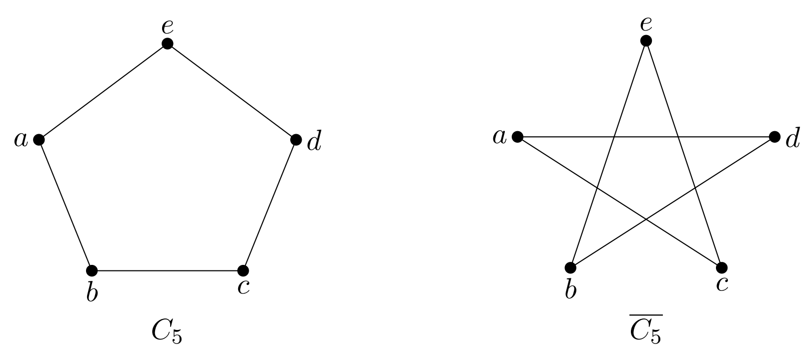


Figure 1.12: C_5, $\overline{C_5}$

Definition 1.1.37. Let G and H be simple graphs. We say G is *isomorphic* to H if there exists a bijection $f : V(G) \to V(H)$ such that for any pair of vertices u and v in $V(G)$, $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$. For example, C_5 is isomorphic to $\overline{C_5}$.

Definition 1.1.38. Given a fixed graph H, an *H-free* graph is a graph such that it does not contain any induced subgraph which is isomorphic to H. For instance, a triangle-free graph is a graph without any triangles. Note that any tripartite graph is K_4-free.

1.2 Matching

Definition 1.2.1. A *matching* in an undirected simple graph is a set of edges which do not have any common endpoint. The vertices incident with the edges in a matching are *saturated* by the matching and those vertices not incident by the edges in the matching are *unsaturated*. If every vertex of a graph is saturated by a matching, then we say this matching is a *complete matching*. A *maximal matching* is a matching such that adding any edge not in the matching will make the new set of edges no longer a matching. A *maximum matching* of a graph that has the largest possible size of a matching in this graph.

Example 1.2.2. The graphs below are examples of a maximal matching (blue) and a maximum matching (red) in the same graph.

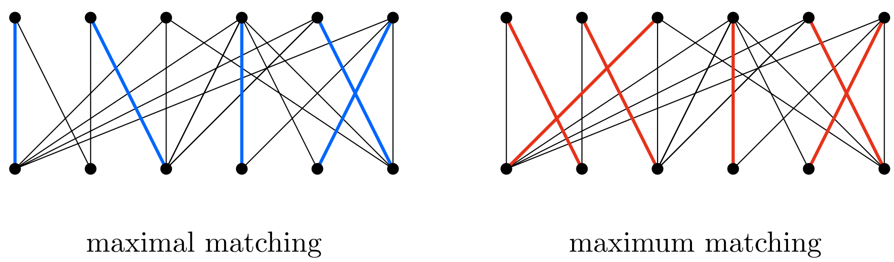


Figure 1.13: Example of maximal and maximum matching

Note that a maximum matching is a maximal matching but a maximal matching does not always have maximum size. We can see from the example above that the maximal matching has size 5 while the maximum matching has size 6.

1.3 Triangles in Graphs

Starting from this section, we will define some terms and notation we will use in this thesis, but these are not used consistently by other authors. In [HK98] pairwise edge-disjoint triangles are described as "independent", but in [HKT12] these are called a "triangle packing". The term "packing" is also used for more general vertex sets, but since we will only focus on triangles in graphs, we abbreviate the term as "packing" to indicate a "triangle packing" in this thesis. Similarly, "triangle edge cover" was used in [HKT12], but to avoid confusion with the more common definition of edge cover given in Definition 1.1.24, we will follow the terminology in [LBT12] and call this a "T-transversal". The term "transversal" is more frequently used in hypergraphs.

Definition 1.3.1. Let T be the set of all triangles in a simple graph G. A subset $T' \subseteq T$ is *independent* if any two triangles in T' do not share any common vertex. We say T' is a *packing* if any two triangles in T' do not share any edge, or equivalently all triangles in T' are pairwise edge-disjoint.

Definition 1.3.2. If $T_m \subseteq T$ is a packing such that $|T_m| \geq |T'|$ for all the packings $T' \subseteq T$, then we say T_m is a *maximum packing* and denote $|T_m| = \nu_{\triangle}(G)$.

Definition 1.3.3. A *T-transversal* is a subset $E' \subseteq E$ such that every element in T contains at least one edge from E'.

Definition 1.3.4. If $E_m \subseteq E$ is a T-transversal in a simple graph G with $|E_m| \leq |E'|$ for all T-transversals E', then we say E_m is a *minimum T-transversal* and denote $|E_m| = \tau_{\triangle}(G)$.

Example 1.3.5.

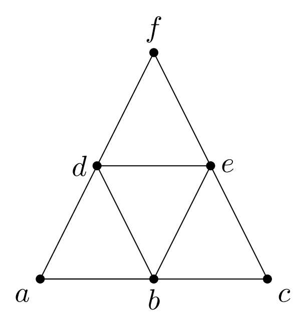


Figure 1.14: Example of a T-transversal

From the graph above we can see there are four triangles in total. The maximum packing is: $\{abd, bce, def\}$, so $\nu_{\triangle}(G) = 3$, and we can cover the four triangles by taking a minimum T-transversal: $\{bd, be, de\}$, so $\tau_{\triangle}(G) = 3$.

Fact 1.3.6. *For any simple graph G:*

$$\nu_{\triangle}(G) \leq \tau_{\triangle}(G) \leq 3\nu_{\triangle}(G).$$

Proof. Assume there exists a maximum packing whose size is greater than $\tau_{\triangle}$, then there will be at least two triangles in this maximum packing will share an edge in a minimum T-transversal and this contradicts the definition of the maximum packing. Thus, $\nu_{\triangle}(G) \leq \tau_{\triangle}(G)$ is always true. The set of all the edges of the triangles in a maximum packing must contain at least one edge of all triangles in G, or else there must exist a bigger packing, which contradicts the definition. So there exists an upper bound $\tau_{\triangle}(G) \leq 3\nu_{\triangle}(G)$. Overall, these give the fact above. □

Definition 1.3.7. Let G be a simple graph, the *triangle graph* of G, denoted as $T(G)$, is the graph with vertices representing the triangles of G, and two vertices of $T(G)$ are adjacent if and only if the corresponding triangles of G share an edge.

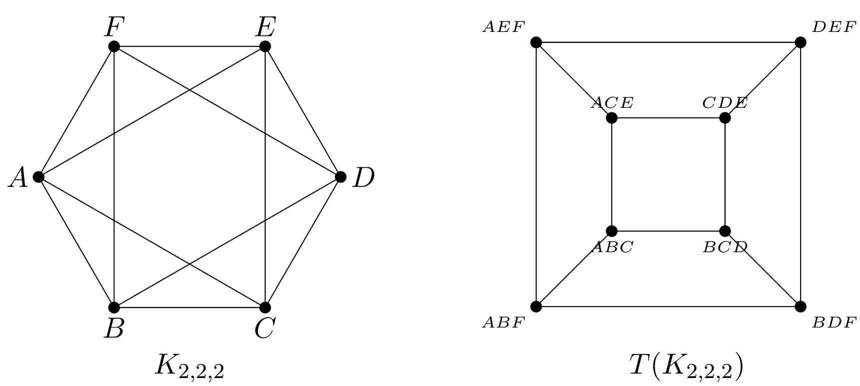


Figure 1.15: Example of triangle graph for $K_{2,2,2}$

Remark 1.3.8. Any graph has a unique triangle graph, but for any given triangle graph it may not always correspond to a unique original graph.

Example 1.3.9. If we are given K_4 as a triangle graph, it has the following two possible original graphs:

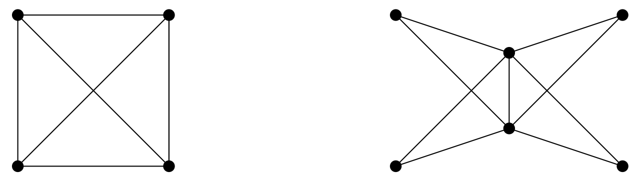


Figure 1.16: Example original graphs of triangle graph K_4

1.4 Hypergraph

Definition 1.4.1. A *hypergraph* H is a generalization of a graph in which the edge set is a subset of the power set of vertices. These edges are called *hyperedges*.

Example 1.4.2. The graph below is an example of a hypergraph with vertex set $V := \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$, and edge set $E :=$ $\{e_1, e_2, e_3, e_4, e_5\} = \Big\{\{v_2, v_4, v_5, v_8\}, \{v_8\}, \{v_6, v_7\}, \{v_1, v_8, v_9\}, \{v_7, v_9\}\Big\}$

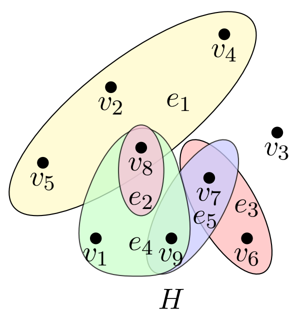


Figure 1.17: Example of Hypergraph

Definition 1.4.3. Let $H = (V, E)$ be a hypergraph, if every element in $E(H)$ has exactly the same size k, then we say H is a k-uniform hypergraph.

Definition 1.4.4. Let H be a hypergraph. The maximum number of disjoint hyperedges is denoted by $\nu(H)$. A *transversal* in a hypergraph is a vertex set which contains at least one vertex from each hyperedge. The minimum size of all transversals is denoted by $\tau(H)$.

Chapter 2

Hall's Marriage Theorem and Extensions in Tripartite Graphs

In this chapter, we will introduce some classical results in concerning bipartite graph: the König-Egerváry Theorem, and its relation to Hall's Marriage Theorem. Then we will mention some generalizations of Hall's Theorem in hypergraphs, and then move on to our work in tripartite graphs.

2.1 Bipartite Graphs and Matchings

In 1931, Dénes König proved that in any bipartite graph, the size of a minimum vertex cover and the maximum matching size are equal. In the same year, coincidentally, Jenö Egerváry proved a more general result independently in weighted graphs [BLW86]. This theorem is now often known as the König and Egerváry Theorem.

Theorem 2.1.1. *König-Egerváry Theorem (K-E Theorem)*
In any bipartite graph, the number of edges in a maximum matching equals the number of vertices of a minimum vertex cover.

In 1935, Philip Hall proved his theorem (known as Hall's Marriage Theorem), which states a necessary and sufficient condition for the existence of a maximum matching that saturates one side of a bipartite graph.

Theorem 2.1.2. *Hall's Matching Theorem*
Let G be a finite bipartite graph with parts X and Y. Let S be a subset of X and $N(S)$ be the set of all the neighbours of vertices in a set S. There exists a matching in G that saturates every vertex in X

if and only if every subset S of X satisfies the following condition:

$$|N(S)| \geq |S|.$$

Example 2.1.3. The graph below is a bipartite graph whose two parts are X, Y.

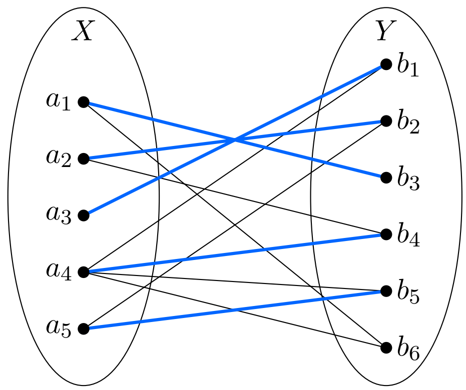


Figure 2.1: Example of a matching of a bipartite graph saturating X

We can use the K-E Theorem to provide a quick proof of Hall's Marriage Theorem as follows:

Proof. Assume $G[X, Y]$ is a bipartite graph, $|S| \leq |N(S)|$ for all subset S in X.
Suppose there exists a vertex cover $W = A \cup B$ where $A \subset Y$ and $B \subset X$ such that $|W| < |X|$. It then follows that $|X \backslash B| > |A|$. But since W is a vertex cover, we have:

$$N(X \backslash B) \subseteq A \Longrightarrow |N(X \backslash B)| \leq |A|.$$

Therefore $|X \backslash B| > |N(X \backslash B)|$, which contradicts our assumption that $|S| \leq |N(S)|$ for all subsets of X. So we obtain that X is a minimum vertex cover. By the K-E Theorem, G has a maximum matching whose size is $|X|$, and by definition of matching, this maximum matching must saturate every vertex in X.

Now suppose there exists a matching M that saturates every vertex in X. Since G is bipartite, X must cover all edges in G. From the assumption that M saturates X, it follows that M is a maximum matching in G, and we also have $|M| = |X|$. Therefore, X is a minimum vertex cover, which is equivalent to $|N(S)| \geq |S| \quad \forall S \subseteq X$. $\square$

Hall's theorem has been used for many real-life matching problems, and applied in combinatorial problems such as creating Latin squares [Bri], and also in group theory [BW09]. We can see that Hall's theorem is a special case of the K-E Theorem, and it determines an important bijection condition for min-max equality in bipartite graphs: there exists an X-saturated matching if and only if X is a minimum vertex cover. So we were wondering if a similar property will still hold in tripartite graphs, which will be introduced in the next section.

2.2 Tripartite Case Extension

In bipartite graphs, an edge can be considered as a pair of mutually adjacent vertices from both parts. Now we are considering tripartite graphs, a similar idea in tripartite graph leads to a set of mutually adjacent vertices from all three parts, which is a triangle.

When trying to extend Hall's theorem to tripartite graphs, we first consider the following idea:

Statement 1: *For any tripartite graph, if one of the vertex parts has the minimum number of vertices required to cover all the triangles in this graph, then the size of this part is equal to the maximum size of a set of independent triangles in this graph.*

But it is very easy to find a counterexample for this statement:

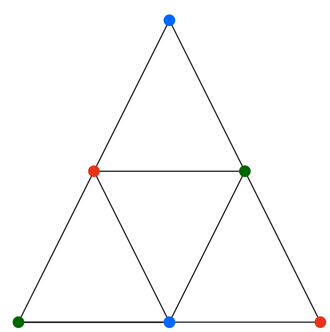


Figure 2.2: Counterexample of statement 1

Vertex colours indicate the vertex partition. If we take any two vertices of the same colour, we can cover all four triangles in the graph. However, we can find only one independent triangle in the graph above.

So it does not look like a promising approach to consider the equality between the size of minimum vertex cover of triangles and maximum independent triangle set for general tripartite graphs.

Since a triangle has three edges, if we have a tripartite graph G then we can define a 3-uniform hypergraph H whose vertices correspond to edges in G, and whose hyperedges correspond to the three edges in a triangle in G.

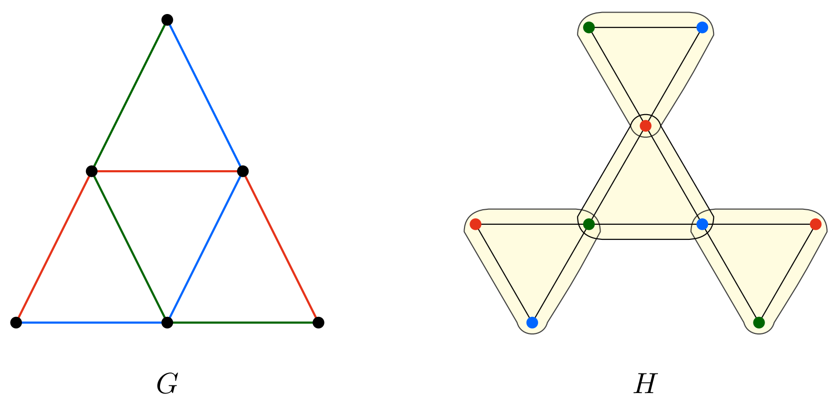


Figure 2.3: Example of a tripartite graph G and the corresponding hypergraph H

We can see there is a "matching" of three disjoint hyperedges in H, and three vertices that cover all the hyperedges in H, so the size of a minimum vertex cover is equal to the maximum size of a matching of disjoint hyperedges in H. These correspond to a minimum T-transversal and a maximum packing in G. Instead, we can consider the size of a minimum number of edges covering all the triangles and the size of a maximum packing, and we will get the following statement:

Statement 2: *For any tripartite graph G, if one side is a minimum T-transversal then $\tau_\triangle(G) = \nu_\triangle(G)$.*

However, this statement is still false in general. Below is the smallest counterexample, which is found in [HK98]:

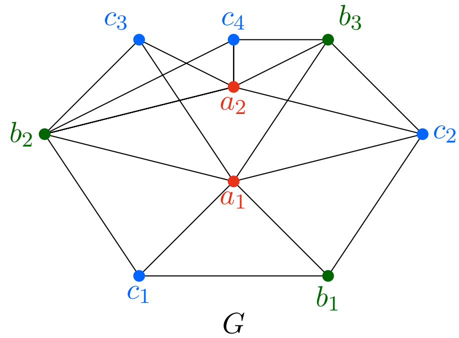


Figure 2.4: Counterexample G

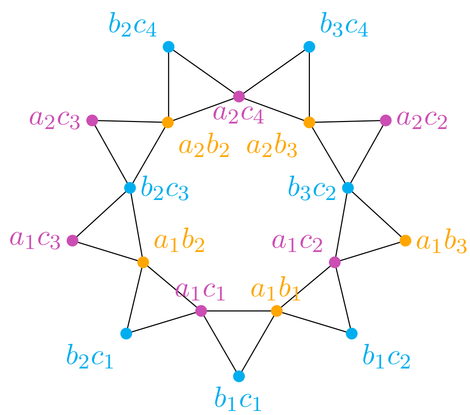


Figure 2.5: Hypergraph H corresponding to G

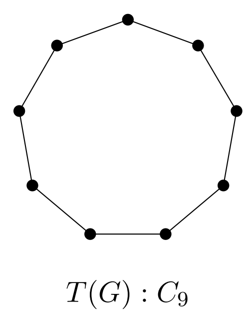


Figure 2.6: Triangle graph $T(G)$

The graph G above has nine triangles in total, and we can see that every edge is contained by at most two triangles, so we will need at least five edges to cover all the triangles in G. The edge set $\{a_1b_1, a_1b_2, a_1b_3, a_2b_1, a_2b_2\}$ has size 5 which is one side of G and hence it is a minimum T-transversal. So G clearly satisfies the condition, but the maximum packing size is 4. To see this, note that finding the maximum packing in G and finding the maximum size of an independent vertex set in $T(G)$ are equivalent. Since $T(G)$ is C_9, a maximum set of independent vertex set cannot consist with two consecutive vertices in this cycle, so the maximum size of an independent set is 4. Therefore, the maximum packing size of G is 4.

The smallest size of counterexample is significantly larger than the smallest counterexample of statement 1, which suggests that the hypergraph point of view might be more promising.

If we consider a bipartite graph as a special kind of hypergraph, then the K-E Theorem is equivalent to the following statement:

For any 2-uniform 2-partite hypergraph H, $\tau(H) = \nu(H)$.

This is just the case $r = 2$ of a conjecture called "Ryser's Conjecture" [Tuz83]:

Conjecture 2.2.1. *Ryser's Conjecture (1971): If H is a r-uniform r-partite hypergraph, then $\tau(H) \leq (r-1)\nu(H)$.*

This conjecture is still open for $r \geq 4$, but the $r = 3$ case was proved by Ron Aharoni in [Aha01] in 2001, by using a generalized hypergraph version of Hall's Theorem, which was proved in 2000 and published in [AH00]. However, the hypothesis of their theorem is very strong and the kinds of tripartite graphs we will consider in Chapter 3 do not satisfy this condition. Therefore, we turn to look at a similar conjecture in the next section.

2.3 Tuza's Conjecture and Related Studies

In 1981, Tuza conjectured the following:

Conjecture 2.3.1. *Tuza's Conjecture: If G is a graph and H is the 3-uniform hypergraph whose vertices are the edges of triangles in G, then $\tau(H) \leq 2\nu(H)$. [Tuz90]*

This hypergraph version description looks very similar to Ryser's conjecture, and it is equivalent to the statement $\tau_\triangle(G) \leq 2\nu_\triangle(G)$ in the graph G. Note that from Fact 1.3.6 that $\tau_\triangle(G) \leq 3\nu_\triangle(G)$, so Tuza conjectured a smaller upper bound of $\tau_\triangle(G)$ for general graphs. It is a weaker relation compared to $\tau_\triangle(G) = \nu_\triangle(G)$ in Statement 2 in the previous section, but it is not limited to a particular graph class.

From the counterexample Figure 2.4 in Section 2.2, it might seem that for any tripartite graph G, if $T(G)$ is odd-cycle-free, then $\nu_\triangle(G) = \tau_\triangle(G)$. In fact, this is proved in [LBT12] as the following theorem:

Theorem 2.3.2. *If G is a K_4-free graph whose triangle graph $T(G)$ is C_{2k+1}-free for all $k \geq 2$, then $\tau_\triangle(G) = \nu_\triangle(G)$.*

Note that all tripartite graphs are K_4-free, but only some of them are applicable to this theorem. For completeness, we will provide a proof with more detail than [LBT12]. Before proving this theorem, we will introduce the Strong Perfect Graph Theorem which was conjectured by Berge Claude in 1961 [Ber61], and was proved in 2006 by Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The proof can be found in [CRST06].

Theorem 2.3.3. *(Strong Perfect Graph Theorem) A graph is perfect if and only if it is C_{2k+1}-free and $\overline{C_{2k+1}}$-free.*

We will also need the following facts:

Lemma 2.3.4. *For any K_4-free graph G, a clique of size n in $T(G)$ corresponds to an edge in G that is shared by n triangles.*

Proof. Consider three triangles $\triangle_1$, $\triangle_2$, $\triangle_3$ in the graph G with the corresponding vertices t_1, t_2, t_3 in the triangle graph $T(G)$. Suppose t_1, t_2, t_3 are in a clique, so they form a triangle in $T(G)$. Let u_1, u_2, u_3 be the three vertices of $\triangle_1$ and u_1, u_2, u_3' be the three vertices of $\triangle_2$ with shared common edge u_1u_2. Now assume $\triangle_3$ does not contain the edge u_1u_2. Since t_1, t_3 and t_2, t_3 are adjacent, we have the vertices of $\triangle_3$ are either (u_1, u_3, u_3') or (u_2, u_3, u_3'). Then u_1, u_2, u_3, u_3' will form a K_4 in G, which contradicts that G was assumed to be K_4-free. □

Fact 2.3.5. *For any integer $k > 2$, $\overline{C_{2k+1}}$ contains two cliques sharing an edge.*

Example 2.3.6.

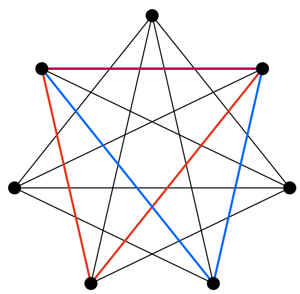


Figure 2.7: Example of Fact 2.3.5 in $\overline{C_7}$

The graph above has two cliques coloured by red and blue, and the shared edge of the two cliques is coloured in purple. For any larger k, $\overline{C_{2k+1}}$ contains $\overline{C_7}$ as an induced subgraph, so Fact 2.3.5 follows immediately from this example.

2.3.1 Proof of Theorem 2.3.2

Proof. Assume G is K_4-free and $T(G)$ is odd cycle-free. By definition, the vertices in $T(G)$ corresponding to a maximum packing of G are independent, so in $\overline{T(G)}$ they are all adjacent to each other, which forms a clique in $\overline{T(G)}$. So $\nu_{\triangle}(G)$ is equal to the maximum clique number of $\overline{T(G)}$, which is:

$$\nu_{\triangle}(G) = \omega(\overline{T(G)}).$$

By Lemma 2.3.4, a clique in $T(G)$ corresponds to triangles sharing one edge in G. Thus, the minimum number of cliques to cover all vertices in $T(G)$ equals the minimum number of edges covering all the triangles in G. By definition, $\tau_{\triangle}(G)$ is equal to the size of minimum T-transversal in G. This gives:

$$\theta(T(G)) = \tau_{\triangle}(G).$$

By Fact 1.1.32, we have $\chi(\overline{T(G)}) = \theta(T(G)) = \tau_{\triangle}(G)$.

From above, since $\nu_{\triangle}(G) = \omega(\overline{T(G)})$ and $\tau_{\triangle}(G) = \chi(\overline{T(G)})$, this means that $\nu_{\triangle}(G) = \tau_{\triangle}(G)$ if and only if $\omega(\overline{T(G)}) = \chi(\overline{T(G)})$ for the graph G.

Since we assume $T(G)$ is odd cycle-free and C_5 is isomorphic to $\overline{C_5}$, so $T(G)$ is $\overline{C_5}$-free. From Lemma 2.3.4 and Fact 2.3.5, we can show that $T(G)$ is $\overline{C_{2k+1}}$-free for all $k \geq 2$. If not, then there are two cliques in $T(G)$ sharing more than one vertex, corresponding to two edges in G that are common to more than one triangle in G. This is a contradiction since two triangles cannot share more than one common edge for any simple graph.

So $T(G)$ is $\overline{C_{2k+1}}$-free, and by assumption, $T(G)$ is C_{2k+1}-free. Therefore $\overline{T(G)}$ is also C_{2k+1}-free and $\overline{C_{2k+1}}$-free. Applying Theorem 2.3.3, we obtain that $\overline{T(G)}$ is perfect, which implies:

$$\chi(\overline{T(G)}) = \omega(\overline{T(G)}).$$

Therefore, we obtain:

$$\tau_{\triangle}(G) = \nu_{\triangle}(G).$$

Therefore, if G is K_4-free and $T(G)$ is C_{2k+1}-free, then $\tau_{\triangle}(G) = \nu_{\triangle}(G)$.
□

Note that this theorem only defines a sufficient condition for a tripartite graph to have equality between $\tau_{\triangle}$ and $\nu_{\triangle}$ (its triangle graph is C_{2k+1}-free). We have found some tripartite graphs (shown in Chapter 3 and 4) whose triangle graph contains many induced odd cycles and they still satisfy the above equality. However, these graphs are large and it is also very hard to find a maximum packing in a big graph. So we would like to introduce a method in Chapter 4 for presenting all triangles in any tripartite graph in an easier way.

Chapter 3

Agrawal's Conjecture and Edge $\triangle$-r-Regular Tripartite Graph

In this chapter, we will explain Hiralal Agrawal's conjecture and the relationship between his design structure and edge $\triangle$-r-regular tripartite graphs.

3.1 Symmetric Design and Agrawal's Conjecture

Definition 3.1.1. A *block design* (v, b, r, k, λ) is a family of b subsets (called "blocks") of a set of v points, such that each block contains k points, and each point occurs in r blocks, and any two different points are in precisely λ common blocks.

Definition 3.1.2. A *symmetric (v, k, λ)-design* is a block design containing v points and v blocks, where each block has size k and any two different points are on precisely λ common blocks. [BJL99]

A typical example of symmetric design is (7,3,1)-design. Let the set of 7 points be $\{0, 1, 2, 3, 4, 5, 6\}$ and let the 7 blocks be given by:

$$\{1, 2, 4\}, \{2, 3, 5\}, \{3, 4, 6\}, \{4, 5, 0\}, \{5, 6, 1\}, \{6, 0, 2\}, \{0, 1, 3\}.$$

Note that out of the 7 blocks formed by these points, each point occurs 3 times and each block contains 3 points, and any two different points are in 1 block.

The figure below is a typical way of presenting the (7,3,1)-design, which is also known as the Fano plane. In the Fano plane, a block is represented as a line or a circle containing three points:

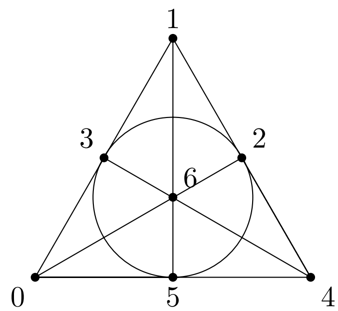


Figure 3.1: Fano plane

Hiralal Agrawal in [Agr66] suggested using symmetric designs in a method for constructing a new structure in statistical design theory. This structure is defined in [McS05] as the following:

Definition 3.1.3. A *triple array* is an $r \times c$ array on v symbols such that no two in one row or in one column and having parameters $(v, k, \lambda_{rr}, \lambda_{cc}, \lambda_{rc} : r \times c)$ such that:
1) Each symbol occurs k times
2) Any two rows having λ_{rr} common symbols.
3) Any two columns share λ_{cc} common symbols
4) Any row and column share λ_{rc} common symbols.

This construction first needs to pick a block B_1 from the symmetric design and use its points to labels the rows of the triple array. Use all the other $v-k$ points which are not in B_1 (equivalently in $\overline{B_1}$) to label the columns. These form a $k \times (v-k)$ array of empty cells where each cell has position (i,j), for some $i \in B_1,\ j \in \overline{B_1}$. If we consider all the other blocks as the set of symbols, then each cell (i,j) is first filled with those symbols representing blocks that do not contain i but do contain j. We then choose exactly one symbol from each cell such that none of these chosen symbols will occur more than once in a row or in a column. Since we start from a symmetric design, this structure turns out to satisfy all the conditions of a triple array.

The structure below is an example formed by the (7,3,1)-design. If we choose to fix the block having points $1, 2, 4$ so that these points label the rows, then $0, 3, 5, 6$ label the columns. The rest of the blocks are the symbols that are allowed to be filled in the labelled cells. We simplify these symbols by giving them a name:

$$B = \{0, 1, 3\} \quad C = \{2, 3, 5\} \quad D = \{3, 4, 6\}$$

$$E = \{4, 5, 0\} \quad F = \{5, 6, 1\} \quad G = \{6, 0, 2\}$$

	0	3	5	6
1	EG	CD	CE	DG
2	BE	BD	EF	DF
4	BG	BC	CF	FG

Table 3.1: Example of Agrawal's structure from (7,3,1)-design

Now we try to choose only one symbol from each cell in Table 3.1 such that no symbol occurs more than once in the same row or column, in order to find a triple array from it. However, the best set of symbols that we pick cannot satisfy all the conditions of a triple array. Below is an example of the partially satisfied triple array from (7,3,1)-design:

	0	3	5	6
1	E	D	C	
2	B		E	DF
4	G	BC	F	G

Table 3.2: Sample partial satisfied triple array of Table 3.1

In other words, Agrawal's construction does not apply in every case. However, this is the only known counterexample from his construction and starting from the next smallest symmetric (11,5,2)-design, we could obtain the following construction:

If we use $\mathbb{Z}_{11}$, take the quadratic residues as the points in the first block, $B_1 = \{1, 3, 4, 5, 9\}$, then non-quadratic residues will be the points labeling the columns: $\overline{B_1} = \{0, 2, 6, 7, 8, 10\}$. Assign the remaining blocks as the following symbols:

$$\begin{aligned} B &= \{2, 4, 5, 6, 10\} & C &= \{3, 5, 6, 7, 0\} \\ D &= \{4, 6, 7, 8, 1\} & E &= \{5, 7, 8, 9, 2\} \\ F &= \{6, 8, 9, 10, 3\} & G &= \{7, 9, 10, 0, 4\} \\ H &= \{8, 10, 0, 1, 5\} & I &= \{9, 0, 1, 2, 6\} \\ J &= \{10, 1, 2, 3, 7\} & K &= \{0, 2, 3, 4, 8\} \end{aligned}$$

The table below shows all the possible symbols that are allowed to be in each position:

	0	2	6	7	8	10
1	CGK	BEK	BCF	CEG	EFK	BFG
3	GHI	BEI	BDI	DEG	DEH	BGH
4	CHI	EIJ	CFI	CEJ	EFH	FHJ
5	GIK	IJK	DFI	DGJ	DFK	FGJ
9	CHK	BJK	BCD	CDJ	DHK	BHJ

Table 3.3: Example of Agrawal's structure from (11,5,2)-design

Now if we pick one symbol from each cell such that all picked symbols occur only once in a row and column, then we can obtain a triple array from this structure. Below is an example triple array obtained from above:

	0	2	6	7	8	10
1	C	B	F	E	K	G
3	I	E	D	G	H	B
4	H	I	C	J	E	F
5	G	K	I	D	F	J
9	K	J	B	C	D	H

Table 3.4: Example triple array (10,5,3,2,3: 5×6) from (11,5,2)-design

These leads to Agrawal's conjecture:

Conjecture 3.1.4. *If there is a symmetric $(v+1, r, \lambda_{cc})$-design with $r - \lambda_{cc}$ >2, then there is a triple array $(v, k, \lambda_{rr}, \lambda_{cc}, \lambda_{rc} : r \times c)$ with $v = r + c - 1$ [NÖ15]*

We can see that the $(7, 3, 1)$-design where $3 - 1 = 2 \not> 2$ is not included in this conjecture. The converse of this conjecture was proved in [PWY05]. One of the infinite families of triple arrays that has been found is named "*Paley Triple Arrays*". While all the previous work about this conjecture is in design theory [Wal14], we found a graphical method to approach this conjecture, which will be introduced in the next section.

3.2 Agrawal's Construction and Tripartite Graph

Consider Agrawal's construction from any symmetric $(v+1, r, \lambda_{cc})$-design where we pick a block B_1 and let W be the remaining v blocks. We can make a tripartite graph whose vertex parts are: B_1 (r vertices), $\overline{B_1}$ ($v+1-r$ vertices) and W (v vertices). We make a complete bipartite graph on B_1 and $\overline{B_1}$, so that these edges represent the cells in the array in Agrawal's construction. Any vertex $i \in B_1$ and a block $A \in W$ are adjacent if $i \notin A$, and any vertex $j \in \overline{B_1}$ is adjacent to a block $A \in W$ if $j \in A$, so that these represent the cells in which the symbol corresponding to A appears in Agrawal's array. Each vertex in W is adjacent to $r - \lambda_{cc}$ vertices in each of the two other vertex sets.

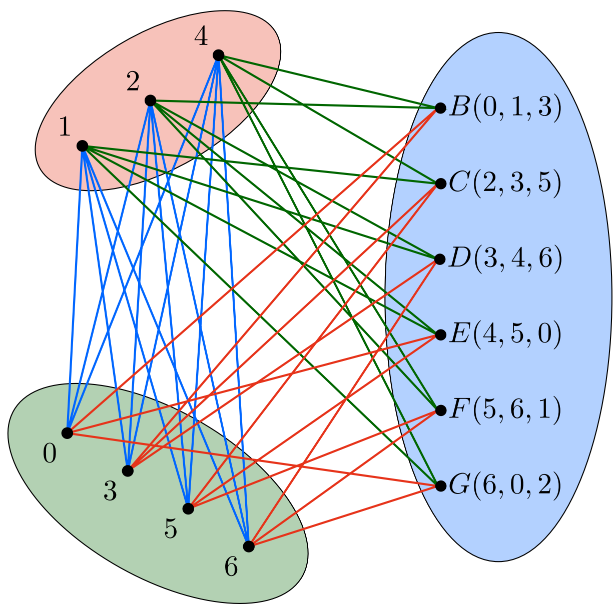


Figure 3.2: Agrawal's construction from (7,3,1)-design in tripartite graph

From Agrawal's construction, we have obtained a special kind of tripartite graph that has the exact same number of edges in each side and such that every edge is shared by the same number of triangles. We define this kind of tripartite graph as the following:

Definition 3.2.1. A tripartite graph G is *edge $\triangle$-r-regular* if every edge in G is shared by exactly r triangles.

Note that a tripartite graph G being edge $\triangle$-r-regular is not equivalent to it being edge regular, since an edge $\triangle$-r-regular graph does not need to be a regular graph. Like the example from above, the vertices from different vertex sets have different degrees, where edge regular graphs are required to be regular graphs. If $G[A, B, C]$ is edge $\triangle$-r-regular, then every edge in G is shared by the same number of triangles, so we obtain that every side has the same number of edges, that is:

$$|E_{AB}| = |E_{AC}| = |E_{BC}|.$$

Let E_T be a T-transversal of an edge $\triangle$-r-regular tripartite graph $G[A, B, C]$. Since every edge is shared by r triangles, so E_T covers at most $|E_T| \cdot r$ triangles. But we know that the total number of triangles in G is $|E_{AB}| \cdot r$. So for any T-transversal, we have:

$$|E_T| \cdot r \geq |E_{AB}| \cdot r \Longrightarrow |E_T| \geq |E_{AB}|$$

This gives every edge side of G is a minimum T-transversal.

It is interesting that any regular bipartite graph satisfies the Hall's matching condition, which is: one part of regular bipartite graph is a minimum vertex cover. And now we found a similar result in tripartite graph as any edge $\triangle$-r-regular tripartite graph always satisfies that one side is a minimum T-transversal.

To make a triple array, we need to pick one symbol from each cell such that every picked symbol only appears once in a row and column of all positions. From the perspective of the graph that we constructed from Agrawal's method, a triple array is equivalent to picking one triangle from each edge of the complete bipartite side such that all these picked triangles are pairwise edge-disjoint. If any two of these triangles are not edge-disjoint, then the corresponding symbols will appear more than once in a column or in a row. This is exactly the same as finding a (maximum) packing that saturates a side of the tripartite graph, so that we can have a maximum packing whose size is equal to the size of a minimum T-transversal. So we can generalize the Agrawal's conjecture in graph perspective as the following:

Conjecture 3.2.2. *For any edge $\triangle$-r-regular tripartite graph G ($r \geq 3$), $\tau_\triangle(G) = \nu_\triangle(G)$.*

Our approach to this conjecture will be introduced in the next chapter.

Chapter 4

Triangle Presentation of Tripartite Graphs

In this chapter, we will introduce a method for listing all triangles for any tripartite graph.

4.1 Method of Listing Triangles

A common method of studying triangles of a given graph is by studying its triangle graph. However, the triangle graph does not necessarily contain the information that it came from a tripartite graph. Likewise, it is also hard to determine whether or not a given graph is the triangle graph of a tripartite graph. So we need an easier way to present all of the triangles in any tripartite graph in order to study them.

Let $G[A, B, C]$ be any undirected finite tripartite graph with vertex partition sets: $A := \{a_i\}$, $B := \{b_j\}$ and $C := \{c_k\}$ and the edge subsets: E_{AB}, E_{AC} and E_{BC}. We assumed E_{AB} has the fewest number of edges. Without loss of generality, we assume every vertex and edge in G is contained in at least one triangle. Since a triangle in G contains one vertex from each part, we can name any triangle in G unambiguously as: $a_i b_j c_k$ and clearly it contains three edges from different edge sets: $a_i b_j$, $a_i c_k$ and $b_j c_k$. If we make an array, then we can use all the vertices of A to label the columns and all the vertices of B to label the rows according to the numerical order of their subscripts in the set. If there is a triangle $a_i b_j c_k$ in G, then we can put c_k in this cell.

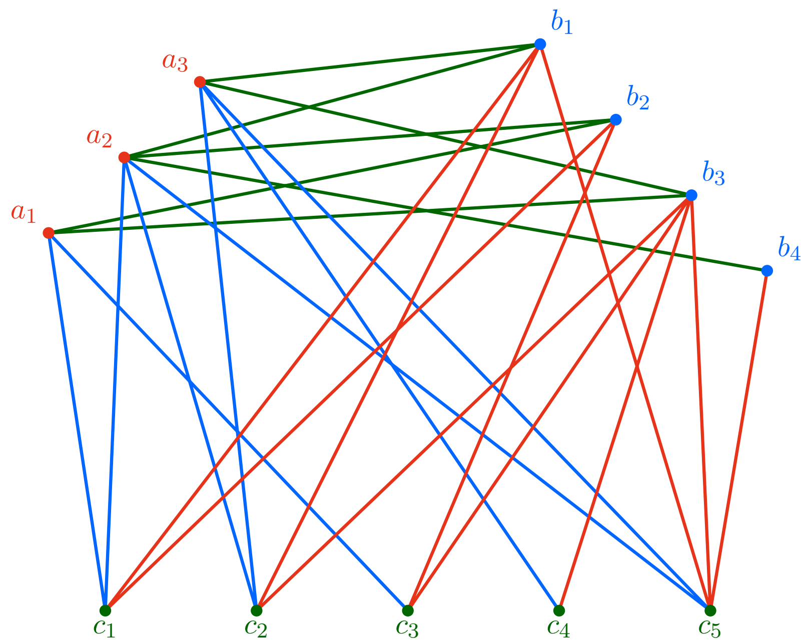


Figure 4.1: Example graph G

For example, the graph G above is a tripartite graph whose all edges and vertices are contained by at least one triangle. The array of G below shows all the triangles in G.

	a_1	a_2	a_3
b_1	$\varnothing$	$c_1c_2c_5$	c_2c_5
b_2	c_1c_3	c_1	$\varnothing$
b_3	c_3	$\varnothing$	$c_2c_4c_5$
b_4	$\varnothing$	c_5	$\varnothing$

Table 4.1: Array of triangles of G

For the non-adjacent pairs of vertices between A and B, we put "$\varnothing$" in those cells. Both c_1 and c_3 appear in the a_1 column and b_2 row. It shows the existence of the two triangles: $a_1b_2c_1$ and $a_1b_2c_3$, which clearly presents the two triangles are sharing the edge a_1b_2. So if a cell of a_i and b_j contains n vertices from C, then it is indicating there are n triangles sharing the edge a_ib_j. If there are some triangles sharing an edge from E_{AC} or E_{BC}, then we can see this from repeated values of subscripts in a column or in a row. For example, c_3 appears in two cells of the a_1 column. It shows that there are two adjacent triangles $a_1b_2c_3$ and $a_1b_3c_3$ sharing the edge a_1c_3. So we can conclude that if any c_k appears λ times in the a_i column, then there are λ triangles sharing the edge a_ic_k. Similarly, if c_k appears δ times in the same b_j row, then this indicates δ triangles sharing the edge b_jc_k.

To simplify the presentation of the array of triangles, we can present all the triangles of G by just listing the value of the subscripts of vertices in C in the cells of the array, and call the result the "Triangle Array Representation" of G, denoted by M. We denote the cell in the a_i column and b_j row position by $M_{i,j}$.

∅	125	25
13	1	∅
3	∅	245
∅	5	∅

Table 4.2: Triangle array representation of G

From M, we can see that "1" appears in $M_{1,2}$ and $M_{2,1}$ indicating the existence of the four edges: a_1c_1, b_2c_1 and a_2c_1, b_1c_1. Since in the example of Table 4.2, a_2 and b_2 are adjacent, $a_2b_2c_1$ is a triangle. Hence, "1" appears in $M_{2,2}$. And a_1, b_1 are not adjacent, so "1" cannot occur in $M_{1,1}$. We name those cells having "∅" as *non-existing cells* and these cells having entries as *existing cells*. Then we can obtain the following fact in general:

Fact 4.1.1. *If value "k" appears in both M_{i_1,j_1} and M_{i_2,j_2} then "k" must also appear in M_{i_2,j_1} and M_{i_1,j_2} as long as they are existing cells of M.*

If an M does not conflict with Fact 4.1.1 for all k, then we can always find an undirected tripartite graph G such that every edge in G is contained in at least one triangle. Essentially, M contains all of the information in $T(G)$ of any undirected tripartite graph G, but compared to $T(G)$, M is easier to visualize for any G which has large size, and preserves the tripartite structure of G.

4.2 Minimum T-Transversal in Triangle Array Representation

In the construction of the triangle array representation of G, the existing cells of M represent the edges of the smallest side of G. In Section 2.2 statement 2, we considered the condition that the smallest side of a tripartite graph is a minimum T-transversal. So in this section, we find an equivalent condition for the smallest side E_{AB} to be a minimum T-transversal, and how it appears in the triangle array representation.

It follows from the proof of Hall's Theorem in Chapter 2 that for any bipartite graph $G[X, Y]$:

$$X \text{ is a minimum vertex cover} \Longleftrightarrow \forall S \subseteq X,\ |N(S)| \geq |S|.$$

Similarly, in a tripartite graph $G[A, B, C]$, if E_{AB} is a minimum T-transversal, then for any subset S of E_{AB}, the set of triangles covered by S cannot be covered by a set of edges from the other two sides whose size is less than $|S|$. Denote $Y_S \subseteq (E_{AC} \cup E_{BC})$ to be a minimum set of edges that covers all the triangles which are covered by $S \subseteq E_{AB}$, and then we can obtain the following theorem:

Theorem 4.2.1. *For any tripartite graph $G[A, B, C]$:*

$$E_{AB} \textit{ is a minimum } T\textit{-transversal} \Longleftrightarrow \forall S \subseteq E_{AB},\ |Y_S| \geq |S|.$$

Proof. Assume every $S \subseteq M_{AB}$ satisfies the condition: $|Y_S| \geq |S|$.

Let $W = S_1 \cup W_{AB}$ be a T-transversal, where $W_{AB} \subset E_{AB}$ and $S_1 \subset (E_{AC} \cup E_{BC})$. Suppose $|W| < |E_{AB}|$, then $|W| - |W_{AB}| = |S_1| < |E_{AB}| - |W_{AB}|$. Let S be the set $E_{AB} \setminus W_{AB}$. Necessarily, S_1 must cover all the triangles that covered by S. Since Y_S is a minimum set of edges that covers all the triangles which are covered by S, by definition, $|Y_S| \leq |S_1| < |S|$ and this contradicts the assumption. Therefore $|W| \geq |E_{AB}|$, which implies E_{AB} is a minimum T-transversal.

Now assume $|E_{AB}| = \tau_{\triangle}$. Suppose there exists a set $S \subseteq E_{AB}$ such that $|Y_S| < |S|$. Then there exists a smaller sized set of edges that covers the set of triangles covered by S, and $(E_{AB} \setminus S) \cup Y_S$ is a T-transversal which is smaller than E_{AB}, contradicting the minimality of E_{AB}. Hence, for every $S \subseteq E_{AB}$, $|Y_S| \geq |S|$. □

Since the triangle array representation M is a way of listing all the triangles in a tripartite graph $G[A, B, C]$ where E_{AB} is the smallest side, if G satisfies the condition in Theorem 4.2.1, then we say all the existing cells of M represent a minimum T-transversal. Before moving on to the array version of the theorem above, we need to know what the edges and a T-transversal look like in M.

We say that each existing cell *covers* its entries and these cells represent the edges in E_{AB}. Let $M_{AB} = \{M_{ij} \big| \forall a_i b_j \in E_{AB}\}$ be the set of all existing cells of M. If a value k appears in one column a_i of M, this represents the edge $a_i c_k$ in G, which we can imagine as a *straight line* in the column a_i covering all the entries in that column with value k, which we denote by $C_i(k)$. Similarly for the rows, we can picture the $b_j c_k$ edge as a straight line in the row b_j covering entries with value k and denote it as $R_j(k)$. By definition, a T-transversal covers all the triangles of a graph, so a collection of existing cells and straight lines that covers all the entries in M is equivalent to a T-transversal of the original graph G. Clearly, all the existing cells in M cover all the entries in M, so M_{AB} represents a T-transversal. Now we need to find how does the condition in Theorem 4.2.1 act in triangle array representation.

Example 4.2.2.

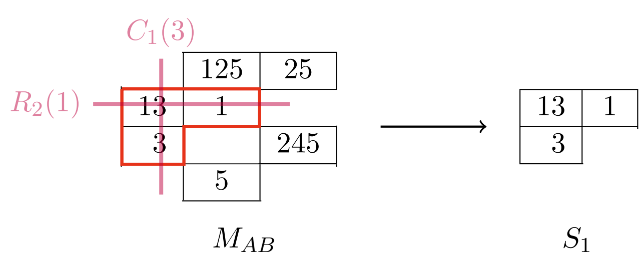


Figure 4.2: Example subset S_1 of M_{AB}

Consider the example triangle array representation from Section 4.1. Take a subset $S_1 := \{M_{1,2}, M_{1,3}, M_{2,2}\}$, which represents three edges $S_{1E} := \{a_1b_2, a_1b_3, a_2b_2\} \subset E_{AB}$ that covers four triangles: $a_1b_2c_1$, $a_1b_2c_3$, $a_1b_3c_3$, $a_2b_2c_1$. We can see these triangles can be covered by the two edges: a_1c_3 and b_2c_1. So if we let $Y_{S_{1E}} = \{a_1c_3, b_2c_1\}$, then $(E_{AB} \setminus S_{1E}) \cup Y_{S_{1E}}$ is a smaller T-transversal compared to E_{AB}. In terms of triangle array representation, if we take $(M_{AB}\backslash S_1)\cup\{C_1(3)\}\cup\{R_2(1)\}$, then we can still cover all the entries in M and it has smaller size than M_{AB}.

So from Theorem 4.2.1 and the correspondence between a graph and its triangle array representation, if M_{AB} represents a minimum T-transversal of G, then the following statement must hold:

Proposition 4.2.3. *Let $G[A, B, C]$ be an undirected simple tripartite graph, M be the corresponding triangle array representation and U_S be a minimum set of straight lines that covers all the entries which are covered by a subset S of M_{AB}, then $|M_{AB}| = \tau_\triangle(G)$ if and only if every $S \subseteq M_{AB}$ satisfies the following condition:*

$$|U_S| \geq |S|.$$

If M_{AB} of a given M satisfies Proposition 4.2.3, then we can say M_{AB} represents a minimum T-transversal of its original graph.

Now we define the "appearance matrix" $A(k)_S$ for each value "k" occurring in a cell of a subset S of M_{AB}, where $A(k)_S$ has the same row and column labeling as M. Each position in $A(k)_S$ has "1" where "k" appears in S of the corresponding position in M, and "0" everywhere else. For instance, the appearance matrices of S_1 in Example 4.2.2 are the following:

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Table 4.3: Example of $A(1)_{S_1}$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Table 4.4: Example of $A(3)_{S_1}$

where $A(2)_{S_1}$, $A(4)_{S_1}$ and $A(5)_{S_1}$ are all zero matrices.

Essentially, a line covering all the "1" in a column or in a row in $A(k)_S$ is equivalent to a straight line that covers all the entries having value "k" in S, which corresponds to an edge from E_{AC} or E_{BC}, respectively. Now we apply a (0,1)-matrix version of König's Theorem [VLWW01]:

Theorem 4.2.4. *The minimum number of lines of (0,1)-matrix that contains all the 1's of the (0,1)-matrix is equal to the maximum number of 1's in the (0,1)-matrix, no two on a line.*

This tells us the smallest number of straight lines to cover all the entries having value "k" in S is equal to the maximum number of entries having value "k" where no two are in the same row or in the same column, and these lines correspond to a minimum set of edges from $E_{AC} \cup E_{BC}$ that covers all the triangles containing c_k in S. For each value k, we define $S(k)$ to be the smallest number of straight lines covering all the entries in S with value "k". Combining Proposition 4.2.3 and Theorem 4.2.4, we can obtain the following:

Corollary 4.2.5. *For a given M of a tripartite graph $G[A, B, C]$, $|M_{AB}| = \tau_\triangle$ if and only if every $S \subseteq M_{AB}$ satisfies the following condition:*

$$\sum_{k=0}^{|C|-1} S(k) \geq |S|.$$

Notice that the minimum set of lines covering all the entries in S with value k is independent for each k. So we can minimize $|U_S|$ by choosing a minimum set of lines covering the entries of each value k in S, and so $|U_S| = \sum_{k=0}^{|C|-1} S(k)$.

Now we want to find a maximum packing of the tripartite graph which size is equal to $\tau_\triangle(G)$. Notice that a packing can be presented in triangle array representation as a selection of entries from M_{AB} where at most one entry has been picked from every cell, and these entries do not appear more than once in a row or a column. If we select the entries with value k separately for each k, then Theorem 4.2.4 guarantees such a selection of maximum size $S(k)$. However, the union of these selections will most likely result in a set of entries with more than one in the same cell. Allowing for this circumstance, we define the term "pseudo-packing" in triangle array representation as the following:

Definition 4.2.6. A *pseudo-packing* P_S of an $S \subseteq M_{AB}$ is a choice of entries from the cells in S such any value "k" appears no more than once in one row or in one column, where each cell can have more than one entry being picked.

Then by Theorem 4.2.4, the maximum number of entries with value k that we can pick in a pseudo-packing is equal to $S(k)$. Combining this result with Corollary 4.2.5 we can also obtain that:

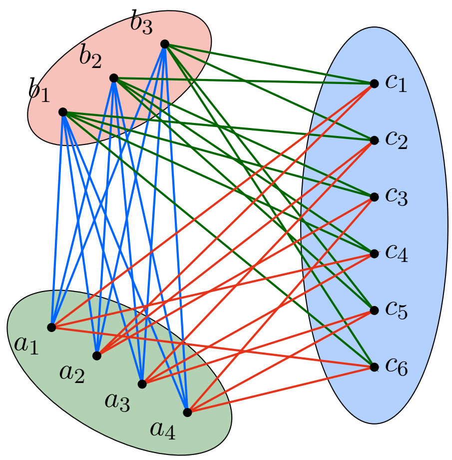
Theorem 4.2.7.

$$M_{AB} \text{ represents a minimum } T\text{-transversal} \\ \Longleftrightarrow \\ \forall S \subseteq M_{AB},\ \exists\ P_S,\ |P_S| = \sum_{k=0}^{|C|-1} S(k) \geq |S|.$$

If an M_{AB} satisfies Proposition 4.2.3, then in particular this theorem guarantees that there exists a pseudo-packing of M_{AB} such that $|P_{M_{AB}}| \geq |M_{AB}|$. Our next step is to make this $P_{M_{AB}}$ less "pseudo" until there is no cell that contains more than one entry of $P_{M_{AB}}$; then we have found a real packing. Our goal is to find an efficient procedure such that we can obtain a maximum packing from this pseudo-packing. If every cell in M_{AB} has exactly one entry in this packing, then equivalently, we obtain a packing of triangles whose size is equal to the minimum T-transversal, which means $\tau_\triangle(G) = \nu_\triangle(G)$. If this can be done whenever G is any edge $\triangle$-r-regular tripartite graph where $r \geq 3$, then we would have proven the Conjecture 3.2.2. We will introduce a technique to help find a maximum packing from a "pseudo-packing" in Chapter 6. For now, we know the statement 2 in Section 2.2 has counterexamples, but Agrawal's conjecture does not have any known counterexample. So we focus on those tripartite graphs constructed from Agrawal's method, which will always give edge $\triangle$-r-regular tripartite graphs.

4.3 Induced Odd Cycles in Triangle Array Representation

From Chapter 3, it is clear that every side of $\triangle$-r-regular tripartite graph from Agrawal's construction has the same number of edges, where one of the sides can be seen as a complete bipartite subgraph. Here we could take the complete bipartite side as the row and the column of the triangle array representation so all the cells are existing cells.



$\longrightarrow$

46	23	24	36
14	13	45	35
16	12	25	56

M

Figure 4.3: Agrawal's construction from (7,3,1)-design in tripartite graph and its array representation

From Chapter 3, we know that every side of an edge $\triangle$-r-regular tripartite graph is a minimum T-transversal. So clearly, for edge $\triangle$-r-regular tripartite graphs, M_{AB} will satisfy the Proposition 4.2.3.

From Theorem 2.3.2, we know that if a triangle graph of a tripartite graph does not contain any induced odd cycle, then it has $\nu_\triangle = \tau_\triangle$. We found the converse of this theorem does not always hold, and we will show some examples of induced odd cycles in the triangle array representation.

An induced odd cycle in a triangle graph is a sequence of an odd number of vertices where any vertex is only adjacent to the consecutive vertices in this sequence. So in triangle array representation, an induced odd cycle will be presented as a cyclic sequence of an odd number of entries which satisfies the following conditions:

1. *Any two consecutive entries in the sequence are either in the same cell or have the same value in a common row or in a common column;*

2. *Any two non-consecutive entries with the same value in the sequence must be in different rows and columns.*
3. *Any two non-consecutive entries with different value cannot be in the same cell.*

We can see from Figure 4.3, there are many induced odd cycles in its triangle graph, the entries highlighted in red below is an induced C_9 in M.

46	23	24	36
14	13	45	35
16	12	25	56

Table 4.5: Example of an induced odd cycle in M

However, for each edge $\triangle$-r-regular tripartite graph from Agrawal's construction where $r \geq 3$, even though we can find many induced odd cycles in its triangle graph, we might still find that the maximum packing has size equal to $\tau_\triangle$. Below is a triangle array representation of graph from (11,5,2)-design.

260	140	125	246	450	156
678	148	138	346	347	167
278	489	258	249	457	579
680	890	358	369	350	569
270	190	123	239	370	179

Table 4.6: Example triangle array representation of graph from (11,5,2)-design and with an example induced C_9

2	1	5	4	0	6
8	4	3	6	7	1
7	8	2	9	4	5
6	0	8	3	5	9
0	9	1	2	3	7

Table 4.7: Example of a maximum packing of Table 4.6

Consider the cases above and Theorem 2.3.2, it gives us a sense that the existence of induced odd cycles in triangle graphs will not be a reason causing the inequality between the $\tau_{\triangle}$ and $\nu_{\triangle}$. Since the smallest counterexample from $(7, 3, 1)$-design is an edge $\triangle$-2-regular tripartite graph, we first focus on this smallest counterexample graph class and try to find some properties which it has and may result in the inequality between $\tau_{\triangle}(G)$ and $\nu_{\triangle}(G)$. The result will be presented in the next chapter.

Chapter 5

Maximum Packing of Edge $\triangle$-2-Regular Tripartite Graphs

This chapter presents our main results on edge $\triangle$-2-regular tripartite graphs.

5.1 Properties of Edge $\triangle$-2-Regular Tripartite Graphs

We provide a result that comes from the smallest counterexample of Agrawal's construction and provide a proof of this result.

Theorem 5.1.1. *If G is edge $\triangle$-2-regular tripartite graph, then $\tau_\triangle(G) = \nu_\triangle(G)$ if and only if $T(G)$ is bipartite.*

Before proving this theorem, we will provide some facts for later proof in Section 5.2.

Lemma 5.1.2. *Let G be any edge $\triangle$-r-regular tripartite graph, then $T(G)$ is k-regular graph where $k = 3 \cdot (r-1)$.*

Proof. By definition, every edge in G is contained in r triangles so every edge of every triangle is also shared by $(r-1)$ other triangles, and hence every triangle in G is adjacent to $3 \cdot (r-1)$ triangles. This means every vertex in $T(G)$ is adjacent to exactly $3 \cdot (r-1)$ neighbours, and therefore $T(G)$ is $3 \cdot (r-1)$-regular. □

Fact 5.1.3. *For any bipartite graph $G[X, Y]$:*

$$\sum_{v \in X} d(v) = \sum_{v \in Y} d(v).$$

Proof. It follows directly from the definition that each edge has one endpoint in X and the other endpoint in Y so the sum of the degrees of all vertices in X is the number of edges in $G[X,Y]$, which is same for the sum of the degrees of all vertices in Y. □

Fact 5.1.4. *For any k-regular bipartite graph $G[X,Y]$ where $k \geq 1$, $|X| = |Y|$.*

Proof. By Fact 5.1.3, we have:

$$|X| \cdot k = \sum_{v \in X} d(v) = \sum_{v \in Y} d(v) = |Y| \cdot k \Longrightarrow |X| = |Y|.$$

□

5.1.1 Proof of Theorem 5.1.1

Proof. Let G be any edge △-2-regular tripartite graph so that every edge in a minimum T-transversal of G is shared by at most two triangles. If there are n triangles in G, then the minimum T-transversal size is greater than or equal to $\frac{n}{2}$.

$$\tau_{\triangle}(G) \geq \frac{n}{2}. \tag{5.1}$$

Since there are n triangles in G, every triangle has three edges, and every edge is shared by exactly two triangles, so $|E(G)| = \frac{3n}{2}$ and each side has the same number of edges, which is $\frac{|E(G)|}{3} = \frac{n}{2}$. So the minimum T-transversal size has to be less or equal to the size of any edge side.

$$\tau_{\triangle}(G) \leq \frac{n}{2}. \tag{5.2}$$

From (5.1) and (5.2), we see $\tau_{\triangle} = \frac{n}{2}$.

"$\Longrightarrow$" Assume $T(G)$ is bipartite. Then by Lemma 5.1.2, $d(v) = 3$ for all vertices in $T(G)$. By Fact 5.1.4 we have:

$$|X| = |Y| = \frac{n}{2}.$$

Since the maximum independent vertex set of $T(G)$ is equivalent to the maximum packing size of G, so we obtain:

$$|X| = \frac{n}{2} = \nu_\triangle(G) \Longrightarrow \nu_\triangle(G) = \tau_\triangle(G).$$

"$\Longleftarrow$" Assume $\tau_\triangle(G) = \nu_\triangle(G)$. From above, we obtain $\tau_\triangle(G) = \nu_\triangle(G) = \frac{n}{2}$ which is equivalent to the size of a maximum independent vertex set in $T(G)$, denoted $|X|$. Since $T(G)$ is 3-regular, we have:

$$\sum_{v \in X} d(v) = 3|X| = \frac{3n}{2}.$$

So X indicates that $T(G)$ has at least $\frac{3n}{2}$ edges. Since $|V(T(G))| = n$, by Fact 1.1.17,

$$|E(T(G))| = \frac{3n}{2}.$$

This means $T(G)$ has exactly $\frac{3n}{2}$ edges. Now suppose $T(G)$ is not bipartite, so by definition, there exists a vertex set which contains some edges both of whose endpoints are not in X. This contradict the above as X contains one endpoint of all edges in $T(G)$. Hence $T(G)$ is bipartite. □

5.2 Orientability of the Surface Formed by an Edge △-2-Regular Tripartite Graph

We will provide some background before moving on to our result.

In graphs, if two triangles are not independent, they are either joined by a vertex or by an edge. Let G be an edge △-2-regular tripartite graph, then the simplicial 2-complex [Mun18] $X(G)$ formed by the union of triangles of G is naturally viewed as a triangulated surface without boundary.

Every triangle has two orientations by cyclically ordering its vertices. For example if we let triangle vertices be "1", "2" and "3", they can be ordered either 123 or 132.

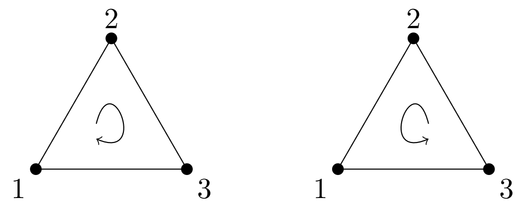


Figure 5.1: Example of triangle orientation

We define the orientability of a surface as the following:

Definition 5.2.1. If every triangle in a triangulated surface can be oriented such that any two triangles sharing an edge have opposite orientation on common edge, then the surface is *orientable*.

A typical example of an orientable surface is "sphere" and a well-known non-orientable surface is "Mobius Strip".

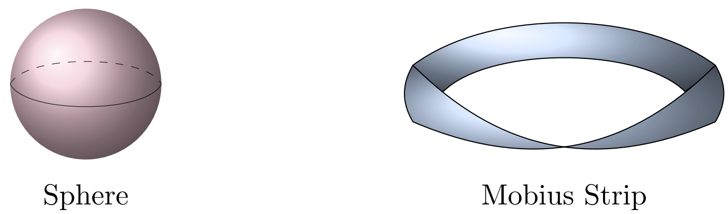


Figure 5.2: Example of a sphere and a mobius strip

If we assign the three colours: red, green and blue to three partitioned vertex sets of a tripartite graph, then we can distinguish triangles in this graph by fix a triangle orientation with vertex colour sequence: red→green→blue ("RGB") so that any orientation of a triangle in this tripartite graph either agrees with "RGB" or disagrees with "RGB". Then for any two edge-joint triangles, one must agree with RGB and the other must disagree with RGB.

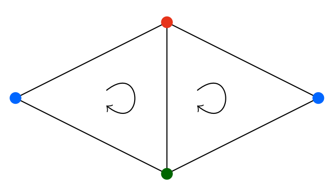


Figure 5.3: Example of two edge-joint triangle with one orientation agrees RGB and the other orientation disagrees with RGB

In any orientation of an orientable surface, we can distinguish the triangles whose orientation agrees or disagrees with RGB. Any two triangles that agree with RGB must be edge-disjoint, so this can be used to partition all the triangles of G into two sets of edge-disjoint triangles. And this is equivalent to all vertices of $T(G)$ can be partitioned into two independent sets $\Longrightarrow T(G)$ is bipartite.

Conversely, suppose $T(G)$ is bipartite, then all adjacent pair of vertices must be in different vertex partition sets. For these corresponding two sets of triangles in G, we can orient the triangles in one set to agree with RGB, and the triangles in the other set to disagree with RGB. So for any two adjacent triangles of G, they must have different orientations. Therefore, $X(G)$ is orientable and we obtain the theorem as below:

Theorem 5.2.2. *The surface $X(G)$ is orientable if and only if $T(G)$ is bipartite.*

Combine the Theorem 5.1.1 and Theorem 5.2.2, we could obtain:

Corollary 5.2.3. *If G is edge $\triangle$-2-regular tripartite graph, then $\tau_{\triangle}(G) = \nu_{\triangle}(G)$ if and only if $X(G)$ is orientable.*

For example, $K_{2,2,2}$ is an edge $\triangle$-2-regular tripartite graph. It is easy to see from below that it forms a closed orientable surface (which called "octahedron").

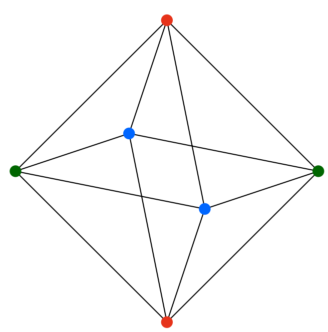


Figure 5.4: Surface formed by triangles in $K_{2,2,2}$

As another example, the graph from Agrawal's construction using (7,3,1)-design turns out to be homeomorphic to a real projective plane, which is a non-orientable surface.

Chapter 6

Pseudo-Packing Technique in Edge $\triangle$-r-Regular Tripartite Graph

In Chapter 3 and 4, we showed that if the size of a maximum packing in a triangle array representation of an edge $\triangle$-r-regular graph is equal to the size of a minimum T-transversal, then we can select exactly one entry from every existing cell such that no two of these entries are having same value in one row and column. We constructed and will show the method for finding a maximum packing from pseudo-packing in the form of triangle array representation that works for general tripartite graph which one side is a minimum T-transversal.

By Theorem 4.2.7, if M_{AB} represents a minimum T-transversal, then we must have a pseudo-packing which size has at least $\tau_{\triangle}$. We know that for any edge $\triangle$-r-regular tripartite graph $G[A, B, C]$, E_{AB} is a minimum T-transversal, thus M_{AB} satisfies the condition of Theorem 4.2.7, hence we can find a pseudo-packing of size $\tau_{\triangle}$ from M_{AB}. We will find a maximum packing from a pseudo-packing and we name this method as: "pseudo-packing technique". We will show this method by showing the procedures on the edge $\triangle$-3-regular tripartite graph obtained from (11,5,2)-design.

For the triangle array representation of a certain graph G, we will find a maximum sized "pseudo-packing" first.

260	14	5			
78		13	46		
	89	2		457	
	0	8	39		56
			2	30	179

Table 6.1: Example pseudo-packing of Table 4.6

By Fact 4.1.1, we know that if entries with value "k" appear in both M_{i_1,j_1} and M_{i_2,j_2}, then the entries with "k" should also appear in M_{i_2,j_1}, M_{i_1,j_2} if they belong to M_{AB}. So we can "switch" these two picked entries from M_{i_1,j_1} and M_{i_2,j_2} to M_{i_2,j_1} and M_{i_1,j_2} in this pseudo-packing.

For example, since both $M_{1,1}$ and $M_{5,5}$ of the pseudo-packing in Table 6.1 have the entry "0", where $M_{1,5}$ and $M_{5,1}$ are empty, we can make a "switch" and reduce the number of empty cells by two for this pseudo-packing.

26	14	5		0	
78		13	46		
	89	2		457	
	0	8	39		56
0			2	3	179

Table 6.2: Example after a switch of Table 6.1

Follow this "switch" process to reduce the number of empty cells until we cannot find any switch to reduce the number of empty cells of this pseudo-packing.

2	1	5	4	0	6
8	4	3	6	7	1
7	8	2		45	9
6	0	8	39		5
0	9	1	2	3	7

Table 6.3: Example after six switches of Table 6.1

Detailed switch steps are shown in Appendix A.1. We can see from above that after the sixth switch, if we do not want to increase the number of empty cells in this pseudo-packing, then we have to look for where after the switch has been made, the number of empty cells does not change.

2	1	5	4	0	6
8	4	3	6	7	1
7	8	2		4	59
6	0	8	39	5	
0	9	1	2	3	7

Table 6.4: Example of the seventh switch of Table 6.1

After this switch, $M_{6,3}$ and $M_{4,4}$ are having the same value entries "9", where $M_{4,3}$ and $M_{6,4}$ are the currently empty cells. So we can make a switch to reduce the number of empty cells by two and hence we obtain a maximum packing.

2	1	5	4	0	6
8	4	3	6	7	1
7	8	2	9	4	5
6	0	8	3	5	9
0	9	1	2	3	7

Table 6.5: Example after eight switch of Table 6.1

In summary, when determining the next action, a switch reducing the number of empty cells by 2 is chosen before one that reduce the number of empty cells by 1, which is chosen before a switch that does not reduce the number of empty cells. However, for those switches reducing the same number of empty cells, we do not have certain criteria to determine which one is better.

We have tested this method in other edge △-r-regular tripartite graphs (For example, see Appendix A.2) and other general tripartite graphs that satisfy one side being a minimum T-transversal. We have found that if a graph does not have the equality $\tau_{\triangle} = \nu_{\triangle}$, then there will always be at least one empty cell and some other cells with more than one entries in the pseudo-packing. As we assume every edge in graph is shared by more than one triangle, every value must occur at least twice in a row or column in the graph's triangle array representation. So if a tripartite graph G satisfies 4.2.7 and has $\tau_{\triangle}(G) > \nu_{\triangle}(G)$, then we will get into endless switches and cannot reduce the number of empty cell by using this method. For example, if we use the same process in the triangle array representation of the graph from Agrawal's construction using (7,3,1)-design, we will perform endless switches not knowing when to stop. For small graphs, we can find the pattern of repetitive switches easily so that we can stop after the repetitive steps occur. However, even though we apply this method only to finite graphs, find repetitive pattern can be extremely hard for large graphs.

If we measure the progress by the number of empty cells in the pseudo-packing, then this technique can be stopped when there is no longer possible to reduce the number of empty cells. This leads us to either of the two cases below without reducing the progress:

(1) Perform endless switches or

(2) Stop as every existing cell is filled

So the result from technique guarantees a maximal packing if we keep one entry only in every non-empty cell from the result. And as far as we have tested, this technique always reaches a maximum packing of G even if $\tau_{\triangle}(G) > \nu_{\triangle}(G)$. Based on our experiment, we conjectured the following statement without a proof:

Conjecture 6.0.1. *If we define the "pseudo-packing technique" as the following:*

First find a maximum sized pseudo-packing. Then make switches with respect to the Fact 4.1.1 with progress measured by reducing number of empty cells and stop this technique when either the following two cases happening:

(1) Perform endless switches or

(2) Stop as every existing cell is filled

Then it is always possible to reach a maximum packing with this method.

Our potential future research for this "pseudo-packing technique" will find efficiency criteria of those switches making the same progress. From the previous example, suppose we choose a different entry to switch from the first step, then our experiment shows that it can take longer process to find a maximum packing through this technique. Therefore, finding the criteria of the efficiency is helpful for prove this conjecture theoretically.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we looked couple results in order to help us to prove Hiralal Agrawal's conjecture.

In Chapter 2, we first focused on generalizing Hall's Theorem in tripartite graph as the graphs obtained from Agrawal's method automatically satisfy Hall's condition in tripartite graph:

If one side of a tripartite graph G is a minimum T-transversal, then $\tau_{\triangle}(G) = \nu_{\triangle}(G)$.

However, we have found counterexample to the above statement.

Later, we focused on finding a stronger condition for obtaining an equivalent relation between $\tau_{\triangle}(G)$ and $\nu_{\triangle}(G)$ by looking at a result from Tuza's conjecture related research:

If G is tripartite graph and $T(G)$ is induced odd cycle-free, then $\tau_{\triangle}(G) = \nu_{\triangle}(G)$.

However, the converse of the theorem above does not hold and we have found counterexamples from graphs obtained through constructions from Agrawal's method.

In Chapter 3, we have defined a type of graph as "edge $\triangle$-r-regular tripartite graph" and generalized the Agrawal's conjecture as the following:

For any edge $\triangle$-r-regular tripartite graph G ($r \geq 3$), $\tau_{\triangle}(G) = \nu_{\triangle}(G)$.

In Chapter 4, we introduced "triangle array representation" for studying edge $\triangle$-r-regular tripartite graph efficiently and we used it to show a counterexample of the converse of the Theorem 2.3.2.

We have proved for any edge $\triangle$-2-regular tripartite graph G, $\tau_{\triangle}(G) = \nu_{\triangle}(G)$ if and only if $T(G)$ is bipartite in Chapter 5 and we use this result to obtain another result: G forms an orientable surface if and only if $T(G)$ is bipartite, which is equivalent to $\tau_\triangle(G) = \nu_\triangle(G)$. It is an interesting direction to look at this question in a topological way.

In Chapter 6, we came up with a "pseudo-packing technique" for finding maximum packing in triangle array representation but we have not found any efficiency criteria for each switch.

7.2 Future Work

From our random general tripartite graph case for finding stopping criteria of the "pseudo-packing procedure", we observed that:

Statement: *For tripartite graph G whose any of its edge is shared by at least three triangles, then $\tau_\triangle(G) = \nu_\triangle(G)$.*

Note that the statement above does not require any side of G to be a minimum T-transversal and we have not found any counterexample for the statement above. This is more like a tripartite graph version of König-Egerváry Theorem. If we can prove this statement, or prove it with a stronger condition: if one side of G is a minimum T-transversal, then we can prove the Conjecture 3.2.2 and hence prove the Hiralal Agrawal's conjecture. But we do not have enough evidence to call it as a conjecture or any appropriate approach for proving it yet.

Below are interesting open questions left from this thesis research:

1. Will the orientability of a triangulated surface be helpful for determining whether or not there exists a maximum packing of size is equal to the minimum edge cover of all triangles in edge $\triangle$-r-regular tripartite graph?

2. Will the "pseudo-packing technique" always produce a maximum packing of a tripartite graph?

3. What are the selection criteria among the switches making the same progress in the "pseudo-packing technique"?

4. Will the following generalized König-Egerváry Theorem be true? That is: For every tripartite graph G whose edges are each in at least three triangles, the size of minimum edge cover of triangles and the size of maximum packing are equal.

Bibliography

[Agr66] Hiralal Agrawal. Some methods of construction of designs for two-way elimination of heterogeneity1. *Journal of the American Statistical Association*, 61(316):1153–1171, 1966. → pages 23

[AH00] Ron Aharoni and Penny Haxell. Hall's theorem for hypergraphs. *Journal of Graph Theory*, 35(2):83–88, 2000. → pages 18

[Aha01] Ron Aharoni. Ryser's conjecture for tripartite 3-graphs. *Combinatorica*, 21(1):1–4, 2001. → pages 18

[Ber61] Claude Berge. Farbung von graphen, deren samtliche bzw. deren ungerade kreise starr sind. *Wissenschaftliche Zeitschrift*, 1961. → pages 19

[BJL99] Thomas Beth, Dieter Jungnickel, and Hanfried Lenz. *Design theory*. Cambridge University Press, 1999. → pages 22

[BLW86] Norman Biggs, E Keith Lloyd, and Robin J Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1986. → pages 13

[BM+76] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976. → pages 1

[Bri] Brilliant.org. Applications of hall's marriage theorem. `https://brilliant.org/wiki/applications-of-hall-marriage-theorem/`. → pages 15

[BW09] John R Britnell and Mark Wildon. Commuting conjugacy classes: an application of hall's marriage theorem to group theory. *Journal of Group Theory*, 12(6):795–802, 2009. → pages 15

[CRST06] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of mathematics*, pages 51–229, 2006. → pages 19

[HK98] Penny E Haxell and Yoshiharu Kohayakawa. Packing and covering triangles in tripartite graphs. *Graphs and Combinatorics*, 14(1):1–10, 1998. → pages 9, 16

[HKT12] Penny Haxell, Alexandr Kostochka, and Stéphan Thomassé. Packing and covering triangles in k 4-free planar graphs. *Graphs and Combinatorics*, 28(5):653–662, 2012. → pages 9, 10

[LBT12] S Aparna Lakshmanan, Cs Bujtás, and Zs Tuza. Small edge sets meeting all triangles of a graph. *Graphs and Combinatorics*, 28(3):381–392, 2012. → pages 10, 19

[McS05] John P McSorley. Double arrays, triple arrays and balanced grids with v= r+c- 1. *Designs, Codes and Cryptography*, 37(2):313–318, 2005. → pages 23

[Mun18] James R Munkres. *Elements of algebraic topology*. CRC Press, 2018. → pages 41

[NÖ15] Tomas Nilson and Lars-Daniel Öhman. Triple arrays and youden squares. *Designs, Codes and Cryptography*, 75(3):429–451, 2015. → pages 25

[PWY05] Donald A Preece, Walter D Wallis, and Joseph L Yucas. Paley triple arrays. *Australasian Journal of Combinatorics*, 33:237, 2005. → pages 25

[Tuz83] Zsolt Tuza. Rysers conjecture on transversals of r-partite hypergraphs. *Ars Combinatoria*, 16:201–209, 1983. → pages 18

[Tuz90] Zsolt Tuza. A conjecture on triangles of graphs. *Graphs and Combinatorics*, 6(4):373–380, 1990. → pages 18

[VLWW01] Jacobus Hendricus Van Lint, Richard Michael Wilson, and Richard Michael Wilson. *A course in combinatorics*. Cambridge university press, 2001. → pages 34

[Wal14] WD Wallis. Triple arrays and related designs. *Discrete Applied Mathematics*, 163:220–236, 2014. → pages 25

[Wes] Douglas B West. Introduction to graph theory. 1996. *Prentiss Hall, Upper Saddle River, NJ.* → pages 1

Appendix

Appendix A

Tables

A.1 Detailed Switch Steps of Edge △-3-regular graph from (11,5,2)-design

26	14	5		0	
78		13	46		
	89	2		457	
	0	8	39		56
0			2	3	179

Table A.1: First switch of Table 6.1

2	14	5		0	6
78		13	46		
	89	2		457	
6	0	8	39		5
0			2	3	179

Table A.2: Second switch of Table 6.1

2	14	5		0	6
78		13	46		
	8	2		457	9
6	0	8	39		5
0	9		2	3	17

Table A.3: Third switch of Table 6.1

2	14	5		0	6
8		13	46	7	
7	8	2		45	9
6	0	8	39		5
0	9		2	3	17

Table A.4: Fourth switch of Table 6.1

2	14	5		0	6
8		3	46	7	1
7	8	2		45	9
6	0	8	39		5
0	9	1	2	3	7

Table A.5: Fifth switch of Table 6.1

2	1	5	4	0	6
8	4	3	6	7	1
7	8	2		45	9
6	0	8	39		5
0	9	1	2	3	7

Table A.6: Sixth switch of Table 6.1

A.2 Triangle Array Representation of graph from (19,9,4)-design

QR of $\mathbb{Z}_{19} = \{1, 4, 9, 16, 6, 17, 11, 7, 5\}$ NQR of $\mathbb{Z}_{19} = \{0, 2, 3, 8, 10, 12, 13, 14, 15, 18\}$

If we take QR of $\mathbb{Z}_{19}$ as the vertex set A and NQR of $\mathbb{Z}_{19}$ as the vertex set B and assign the following subscripts to the vertices of vertex set C:

$$
\begin{array}{llll}
a & = \{2, 5, 10, 17, 7, 18, 12, 8, 6\} & b & = \{3, 6, 11, 18, 8, 0, 13, 9, 7\} \\
c & = \{4, 7, 12, 0, 9, 1, 14, 10, 8\} & d & = \{5, 8, 13, 1, 10, 2, 15, 11, 9\} \\
e & = \{6, 9, 14, 2, 11, 3, 16, 12, 10\} & f & = \{7, 10, 15, 3, 12, 4, 17, 13, 11\} \\
g & = \{8, 11, 16, 4, 13, 5, 18, 14, 12\} & h & = \{9, 12, 17, 5, 14, 6, 0, 15, 13\} \\
i & = \{10, 13, 18, 6, 15, 7, 1, 16, 14\} & j & = \{11, 14, 0, 7, 16, 8, 2, 17, 15\} \\
l & = \{12, 15, 1, 8, 17, 9, 3, 18, 16\} & l & = \{13, 16, 2, 9, 18, 10, 4, 0, 17\} \\
m & = \{14, 17, 3, 10, 0, 11, 5, 1, 18\} & n & = \{15, 18, 4, 11, 1, 12, 6, 2, 0\} \\
o & = \{16, 0, 5, 12, 2, 13, 7, 3, 1\} & p & = \{17, 1, 6, 13, 3, 14, 8, 4, 2\} \\
q & = \{18, 2, 7, 14, 4, 15, 9, 5, 3\} & r & = \{0, 3, 8, 15, 5, 17, 10, 6, 4\}
\end{array}
$$

Then we could obtain the following triangle array representation of an edge $\triangle$-5-regular graph:

bhjlr	bhjmo	jmnor	bchmn	cjlmo	bcnor	chlor	hlmnr	bcjln
aejlq	adejo	ajonp	adnpq	djloq	denoq	alopq	delnp	ejlnp
befqr	bekmo	fmopr	bfmpq	fkmoq	beoqr	kopqr	ekmpr	befkp
abgjr	abdjk	agjpr	abcdp	cdgjk	bcdgr	ackpr	dgkpr	bcjkp
aeflr	adeim	afimr	acdfm	cdflm	cdeir	acilr	delmr	cefil
aefgh	aehko	afgno	acfhn	cfgko	cegno	achko	eghkn	cefkn
bfghl	bdhio	fgiop	bdfhp	dfglo	bdgio	hilop	adhlp	bfilp
eghjq	ehijm	gijmp	chmpq	cgjmq	cegiq	chipq	eghmp	ceijp
fhjqr	dhijk	fijnr	dfhnq	dfjkq	dinqr	hikqr	dhknr	fijkn
abglq	abikm	agimn	abmnq	gklmq	bginq	aiklq	gklmn	bikln

Table A.7: Triangle Array Representation of graph from (19,9,4)-design

j	h	n	c	m	o	r	l	b
l	j	o	d	q	e	a	n	p
q	b	m	p	f	r	o	k	e
r	a	j	b	c	g	p	d	k
e	m	f	a	l	d	c	r	i
a	o	g	f	k	c	h	e	n
g	d	i	h	o	b	l	p	f
h	e	p	m	j	i	q	g	c
f	k	r	n	d	q	i	h	j
b	i	a	q	g	n	k	m	l

Table A.8: Example maximum packing of Table A.7