Star Decompositions of Bipartite Graphs

A thesis

submitted in partial fulfilment

of the requirements for the Degree

of

Masters of Science

at the

University of Waikato

by

J.S. Lim

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

University of Waikato

2015

Abstract

In Chapter 1, we will introduce the definitions and the notations used throughout this thesis. We will also survey some prior research pertaining to graph decompositions, with special emphasis on star-decompositions and decompositions of bipartite graphs. Here we will also introduce some basic algorithms and lemmas that are used in this thesis.

In Chapter 2, we will focus primarily on decomposition of complete bipartite graphs. We will also cover the necessary and sufficient conditions for the decomposition of complete bipartite graphs minus a 1-factor, also known as crown graphs and show that all complete bipartite graphs and crown graphs have a decomposition into stars when certain necessary conditions for the decomposition are met. This is an extension of the results given in "On claw-decomposition of complete graphs and complete bigraphs" by Yamamoto, et. al [38]. We will propose a construction for the decomposition of the graphs.

In Chapter 3, we focus on the decomposition of complete equipartite tripartite graphs. This result is similar to the results of "On Claw-decomposition of complete multipartite graphs" by Ushio and Yamamoto. Our proof is again by construction and we propose how it might extend to equipartite multipartite graphs. We will also discuss the 3-star decomposition of complete tripartite graphs.

In Chapter 4 , we will discuss the star decomposition of r-regular bipartite graphs, with particular emphasis on the decomposition of 4-regular bipartite graphs into 3-stars. We will propose methods to extend our strategies to model the problem as an optimization problem. We will also look into the probabilistic method discussed in "Tree decomposition of Graphs" by Yuster [39] and how we might modify the results of this paper to star decompositions of bipartite graphs.

In Chapter 5, we summarize the findings in this thesis, and discuss the future work and research in star decompositions of bipartite and multipartite graphs.

Acknowledgement

I would like to take this opportunity to express my special appreciation and thank you to my supervisor and advisor Dr. Nicholas Cavenagh, for his support and guidance throughout this project. I would like to thank him especially for his extra time, wisdom, patience, advice and in keeping me focused in this project. He has truly been instrumental in helping me develop the necessary skills to complete this dissertation.

I would like to thank the Faculty and the School of Mathematics in University of Waikato for the opportunity to pursue my post-graduate studies. I appreciate especially the knowledge imparted by the professors and how the academic staff made the post graduate students feel special. I would like to acknowledge my fellow post-graduates in the School of Mathematics and Statistics who have made the office entertaining and homely with the lively discussions and by mutually motivating each other in our respective projects.

I thank my friends in Hamilton, with very special emphasis to my housemates. They have been my family away from home, supporting me and keeping me motivated. I thank them especially for going the extra mile to make my post-graduate life a joy. I also would like to thank my friends back in Malaysia for their encouragement throughout my life as a post-graduate.

Last but absolutely not least, I would also like to thank my family members who have supported me and kept me in their daily prayers. Their constant encouragement and support have been a driving force in completing this dissertation.

I dedicate this dissertation to my grandmother. It has been a difficult year being apart from her in the time of her illness, but her love has sustained me throughout this project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Definitions

Unless stated, all definitions are consistent with "Graph Theory with Applications" [5].

A **graph** is an ordered pair $G = (V, E)$ where V is a non-empty set of **vertices** and E is a set of **edges** which are subsets of V of size 2. The order of the graph $|V|$ is the number of vertices and the graph size $|E|$ is the number of edges in the graph.

In the case of **directed graphs** or **digraphs**, the order of the 2 elements is considered unique and each element of the set E is known as an **arc** or **directed edge**. A **loop** is an edge with the starting and ending vertices equal. We say that the graph contains a **multiple edge** if the graph contains two or more edges joining the same pair of vertices. A vertex is said to be **adjacent** to another vertex if there is an edge between the two vertices. A vertex is said to be **incident** to an edge if the vertex is contained in the edge.

Simple graphs are undirected graphs that do not contain any loops or multiple edges. Thus each edge in a simple graph is a distinct unordered pair of vertices.

From here onwards a graph is assumed to be simple and undirected unless otherwise stated.

A **walk** of length n is a sequence $[v_1, v_2, \ldots, v_{n+1}]$ of vertices, such that $\{v_i, v_{i+1}\}$ is an edge for each $1 \leq i \leq n$. If the edges are all distinct from one another, the walk is called a **trail**. If the both the edges and vertices are all distinct, the walk is called a **path**. A path is denoted by P_n where n is the number of vertices in the path.

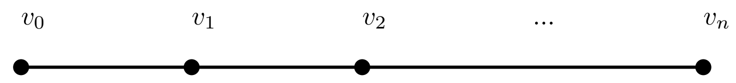


Figure 1.1: Path from v_1 to v_n.

A **circuit** is a non-trivial trail in a graph from a vertex to itself. If all the vertices except for the first vertex and last vertex in the circuit are distinct, the circuit is called a **cycle**. A graph that does not contain any cycles is known as a cycle-free graph. A cycle is denoted by C_n where n is the number of vertices in the cycle.

Formally, let $V = \{v_i : 1 \leq i \leq n\}$ be a set of distinct vertices, and let $E = \{e_i : 1 \leq i \leq n\}$ where $e_i = \{v_i, v_{i+1}\}$ for $1 \leq i \leq n-1$ and $e_n = \{v_n, v_1\}$. Then the graph $C_n = G(V, E)$ is a cycle of length n.

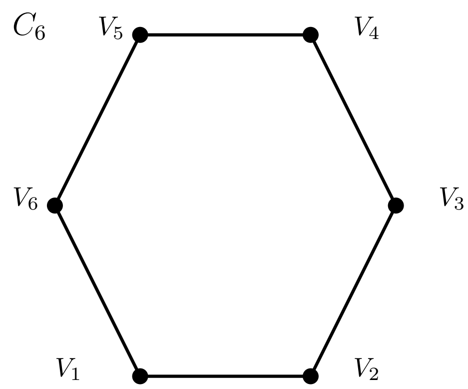


Figure 1.2: Cycle of length 6.

A **connected component** or **component** of a graph is a subgraph such that for every pair of vertices $\{u, v\}$ within the component there exists at least one path from u to v. If the graph consists of exactly one connected component the graph is called a **connected graph**. A **bridge** is an edge such that the removal of the edge results in an increase in the number of components in the graph. If $H_1, H_2, \ldots H_n$ are the components of the graph G then we can also use the notation $G = H_1 \bigcup H_2 \cdots \bigcup H_n$.

A connected graph is said to have an **Eulerian Trail** if there exists a trail such that each edge of the graph is used exactly once. If the trail starts and ends on the same vertex, the graph is said to have an **Eulerian Circuit**. A graph that has an Eulerian Circuit is also said to be **Eulerian**. An Eulerian Circuit exists in a connected graph if and only if every vertex in the graph has even degree.

A graph is said to have a **Hamilton Path** if there exists a path such that each vertex of the graph is visited exactly once. If there exists a cycle such that every vertex of the graph belongs to the cycle, the graph is said to have a **Hamilton Cycle**. Equivalently, the graph is said to be **Hamiltonian**.

The **line graph** $L(G)$ is a graph such that the edge set of G is the vertex set of $L(G)$, and the edge set $E(L(G))$ is such that there is an edge if and only if there is a vertex in common with the corresponding edges in G.

Formally, $V(L(G)) = E(G)$ and $E(G) = \{e_i, e_j\}$ if and only if e_i and e_j share a common vertex in G. Figure 1.3 shows an example graph G and the corresponding line graph. According to Skiena [34], the line graph of an Eulerian graph is both Hamiltonian and Eulerian.

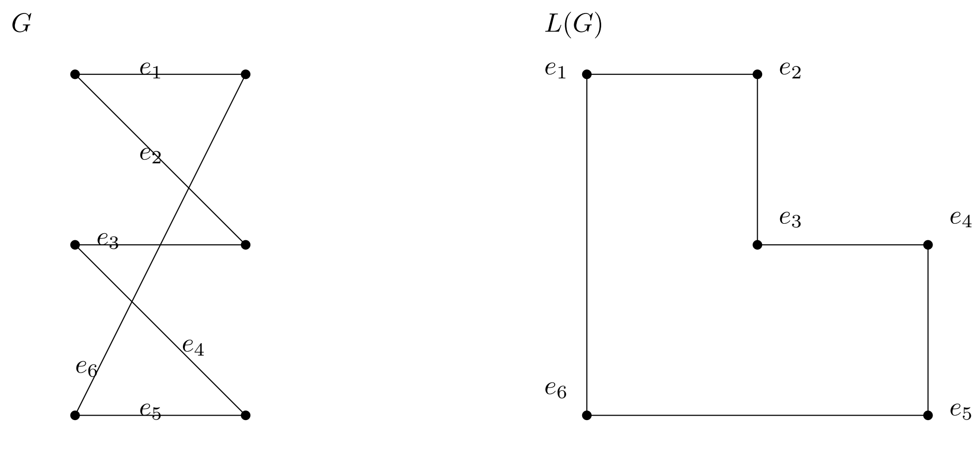
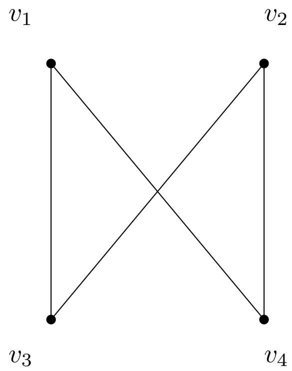


Figure 1.3: Graph G and its Line Graph $L(G)$.

The **degree** $\delta(v)$ of vertex v is the number of edges incident to the vertex v. If every vertex in a graph has the same degree r, the graph is said to be r-**regular**.

A graph H is said to be **isomorphic** to graph G, if there is a bijection $f : V(G) \to V(H)$ such that $\{v, w\} \in E(G)$ if and only if $\{f(v), f(w)\} \in E(H)$.

An **incidence matrix** is an $n \times m$ matrix $B = [b_{ij}]$ where n is the number of vertices and m is the number of edges, subject to the following. If the vertex set $V = \{v_1, v_2, \ldots v_n\}$ and the edge set $E = \{e_1, e_2, \ldots e_m\}$ then b_{ij}=1 if the vertex v_i and edge e_j is incident and $b_{ij} = 0$ otherwise.‘



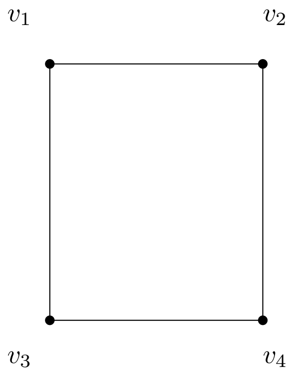
	e_1	e_2	e_3	e_4
v_1	1	1	0	0
v_2	0	0	1	1
v_3	1	0	1	0
v_4	0	1	0	1

Figure 1.4: A graph and its incident matrix.

An **adjacency matrix** is a $n \times n$ matrix $B = [b_{ij}]$ where n is the number of vertices, subject to the following. If the vertex set $V = \{v_1, v_2, \ldots v_n\}$, we let $b_{ij} = 1$ if vertex v_i and vertex v_j are adjacent and $b_{ij} = 0$ otherwise. Observe that for simple graphs, the diagonal of the adjacency matrix is 0. Also observe that for an undirected graph, the adjacency matrix is symmetric.

A **complete graph** is a graph in which every pair of distinct vertices is connected by a unique edge. A complete graph is denoted by K_n where n is the number of vertices in the graph. The edge set of K_n is all the possible edges on the vertex set of G.

Formally G is complete if and only if $E(G) = \{v_i, v_j\}$ where $v_i \in V(G), v_j \in V(G), v_i \neq v_j$.



	v_1	v_2	v_3	v_4
v_1	0	1	1	0
v_2	1	0	0	1
v_3	1	0	0	1
v_4	0	1	1	0

Figure 1.5: A Graph and its adjacency matrix.

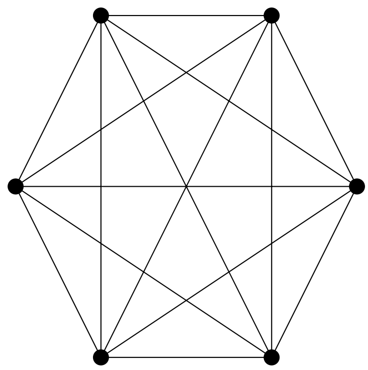


Figure 1.6: Complete graph K_6.

We say that $\overline{G}$ is the **complement** of a graph G such that the vertex set $V(\overline{G}) = V(G)$ and the edge set of $\overline{G}$ consists of all the possible edges that are not present in G. Observe that $E(\overline{G}) + E(G) = E(K_n)$ where $n = |V(G)|$.

Formally $\overline{G}$ is the complement of G if and only if $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{v_i, v_j$ where $v_i \in V(G), v_j \in V(G), v_i \neq v_j$ and $\{v_i, v_j\} \notin E(G)\}$..

A **bipartite graph** (sometimes known as **bigraph**) is a graph in which the vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge is incident with a vertex in V_1 and a vertex in V_2. The sets V_1 and V_2 are known as **partite sets**. Observe that a bipartite graph is either cycle-free or has at least one even cycle. Equivalently, a graph that does not contains an odd cycle is bipartite.

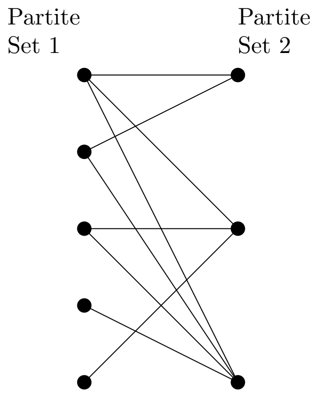


Figure 1.7: Example of a bipartite graph.

A graph G is said to be **multipartite** or m**-partite** if the vertex set V can be partitioned into m disjoint sets $V_1, V_2, \ldots, V_m$ such that every edge of G is incident to vertices from two different partite sets. A multipartite graph is said to be **equipartite** every partite set has an identical size. In the case where $m = 3$ the graph is also known as **tripartite**.

A **complete bipartite graph** is a bipartite graph in which every vertex in V_1 is adjacent to every vertex in V_2. Formally G is a complete bipartite graph if and only if $E(G) = \{v_i, v_j : v_i \in V_1, v_j \in V_2\}$. A complete bipartite graph is denoted by $K_{n,m}$ where $n = |V_1|$ and $m = |V_2|$. We say that a **complete square bipartite graph** is a complete bipartite graph with an equal number of vertices in each partite set.

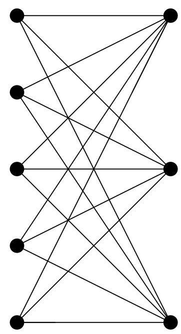


Figure 1.8: The complete bipartite graph $K_{5,3}$.

An r**-regular bipartite graph** is a bipartite graph where every vertex of the bipartite graph has degree r. Observe that an r-regular bipartite graph always has an equal number of vertices in each partite set.

We say that a r**-regular bipartite graph** is "cyclic" if the edges of the graph are induced by ordering the vertices of partite sets U and V and defining an adjacency based on a cyclic difference between the vertices of the partite sets.

We define a **generator** $G_n(D)$ of a r-regular cyclic bipartite graph as the function describing the adjacency between the two partite sets. We call D here the **generator set** where D is of size r. A vertex u in U is adjacent to a vertex v in V if and only if the index of v minus the index of u modulo n is equal to an element in D.

Formally, let $U = \{u_i : 1 \leq i \leq n\}$ and $V = \{v_j : 1 \leq j \leq n\}$ be the partite sets of the bipartite graph. Let $D = \{d_k : 1 \leq k \leq r\}$ where $0 \leq d_j < n$. The vertices u_i and v_j are adjacent if and only if $j = i + d_k (\text{mod } n)$ for some $d_k \in D$. Figure 1.9, is an example of a cyclic 4-regular bipartite graph.

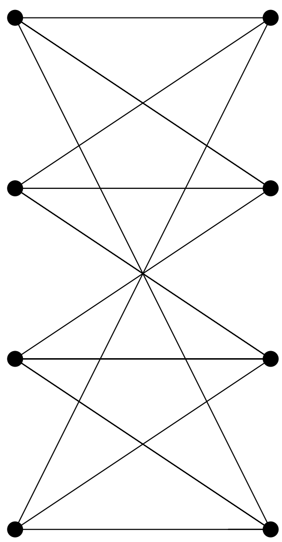


Figure 1.9: 3-Regular Cyclic Bipartite Graph with $n = 4$ and $D = \{0, 1, 3\}$.

A **matching** is a set of edges of a graph such no two edges have a vertex in common. A **perfect matching** is when every vertex of the graph is incident to exactly one edge of the **matching**. A perfect matching is also called a **1-factor** of the graph. A complete bipartite graph $K_{n,n}$ that has a perfect matching removed is known as the **crown graph** of size n [32]. Thus G is a crown graph of size n if $E(G) = \{u_i, v_j : u_i \in V_1, v_j \in V_2, i \neq j\}$ where

$V_1 = \{u_1, u_2, \ldots, u_n\}$ and $V_2 = \{v_1, v_2, \ldots, v_n\}$ are the two partite sets of G. A crown graph of size n is denoted by K_n^0.

A **tree** is a graph in which every pair of vertices is connected by a unique path. The **leaves** of a tree are the vertices of the tree with vertex degree 1. An internal vertex is a vertex of degree at least 2. The **diameter** of a tree is the length of the longest path in the tree. Observe that a tree is cycle-free and thus bipartite.

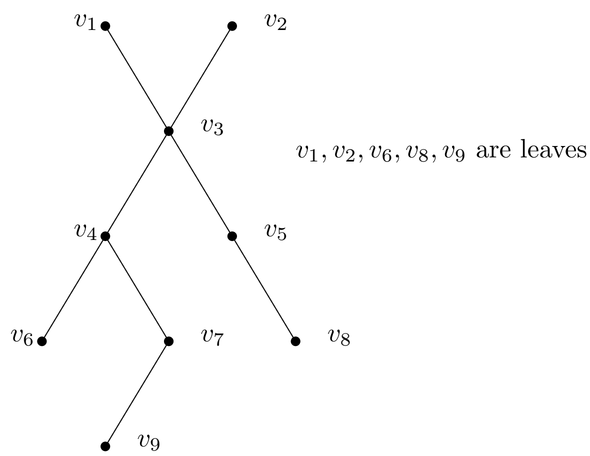


Figure 1.10: Example of a tree.

A k**-star** is a special case of a tree in which there is only one **internal vertex** which is also known as the **center** and k leaves. A k-star is denoted by S_k where k is the number of leaves. A k-star can also be represented as the complete bipartite graph $K_{1,k}$. A 3-star is sometimes known as a **claw**. Observe that the center of S_k has degree k and the leaves of S_k have degree 1. Observe also that the diameter of S_k, where $k \geq 2$ is always two.

We say that the **greatest common divisor** of a graph G, (denoted here as GCD(G) is the greatest common divisor of the degrees of the vertices in G. Observe that when G is a tree or a star, GCD(G)=1.

The graph $H(V', E')$ is a **subgraph** of $G(V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. A graph G is said to **decompose** into $\{G_1, G_2, \ldots G_i\}$ where $G_1, G_2 \ldots G_i$ are subgraphs of G if $E(G)$ has the partition $\{E(G_1), E(G_2), \ldots, E(G_i)\}$. If $G_1, G_2, \ldots G_i$ are all isomorphic to H then we say that there is an H**-Decomposition** of the graph G. Observe that, in order for an H-decomposition

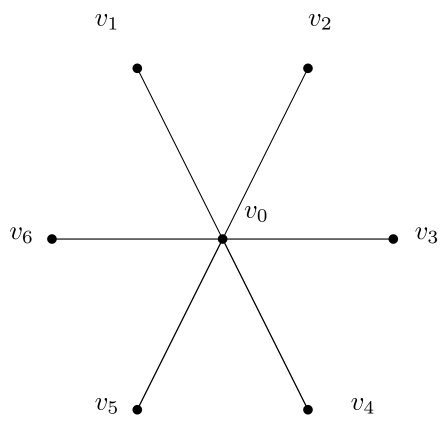


Figure 1.11: Graph S_6; v_0 is the center; $v_1, v_2, v_3, v_4, v_5, v_6$ are the leaves.

to exist, the number of edges in G must be divisible by the numbers of edges in H. Moreover the GCD(G) must also be divisible by GCD(H) [39].

A graph G is said to **factor** into subgraphs $G_1, G_2 \ldots G_i$ if every vertex $V(G)$ has a partition $\{V(G_1), V(G_2), \ldots, V(G_i)\}$. If $G_1, G_2, \ldots, G_i$ are all isomorphic to H, then we say that there is an H**-Factor** in the graph G. If H is the path P_2, then this is equivalently a **1-Factor** of the graph G. An H**-factorization** of a graph G is a decomposition of G into H-Factors.

Figure 1.12 illustrates an example of a P_2-decomposition of a graph, with each coloured lines a copy of a P_2. Figure 1.13 illustrates an example of a P_2-factor of a graph with each bolded lines a P_2 factor, and Figure 1.14 illustrates an example of a C_6-factorization of a graph with the bolded lines a copy of C_6.

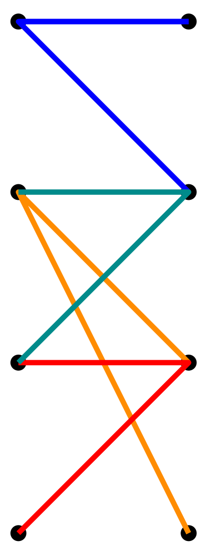


Figure 1.12: P_2-decomposition of a graph.

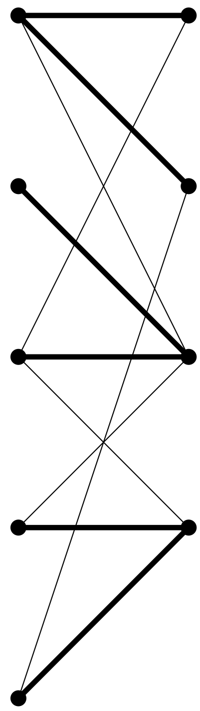


Figure 1.13: P_2-factor of a graph.

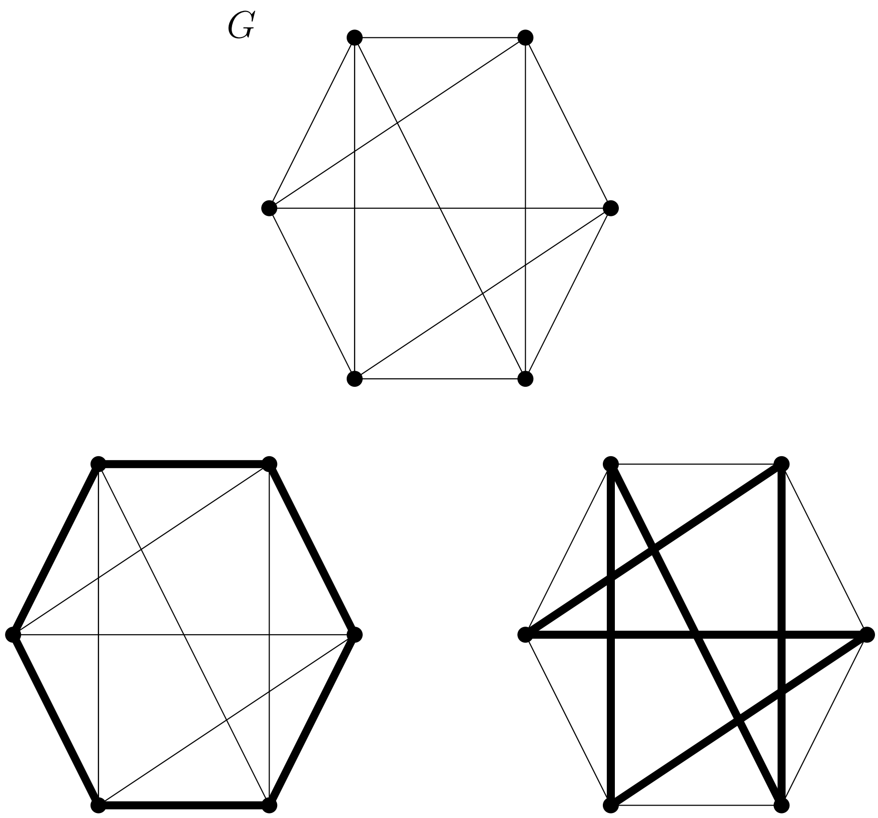


Figure 1.14: C_6-factorization of Graph G.

A **graph product** of G_1 and G_2 is a new graph H where $V(H) = V(G_1) \times V(G_2)$. A special graph product that is used in this thesis is the **lexicographical product**. This was first introduced by Hausdorff according to Imrich and Klavzar [25] [7]. The lexicographical product of G_1 and G_2 is denoted by $G_1 \otimes G_2$. A lexicographical product is a product such that an edge between vertices (u, v) and (x, y) exists if and only if an edge exists between u and x in G_1 or $u = x$ and an edge exists between v and y in G_2. Figure 1.15, shows an example of a lexicographical product.

Formally, if $V(U) = \{u_i : 1 \leq i \leq n\}$ and $V(V) = \{V_j : 1 \leq j \leq m\}$ and $H = U \otimes V$ then $V(H) = \{h_{i,j} : 1 \leq i \leq n, 1 \leq j \leq m\}$ and $E(H) = \{h_{i,j}, h_{k,l}\}$ if and only if $\{u_i, u_k\} \in E(U)$ or $u_i = u_k$ and $\{v_j, v_l\} \in E(V)$.

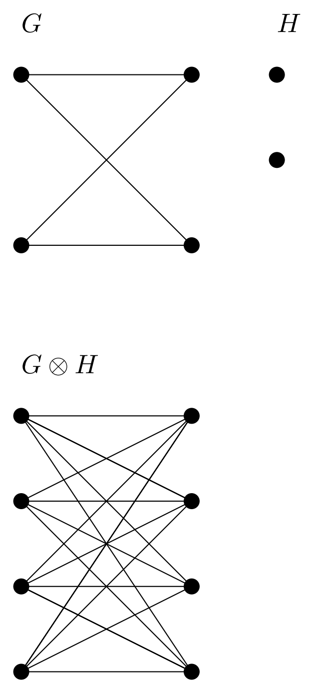


Figure 1.15: Lexicographical product of Graph $G = K_{2,2}$ and $H = \overline{K_2}$.

A **clique** of the graph G is a complete subgraph of G. If the clique is the maximum possible size, the clique said to be the **maximum clique**. Observe that the size of the maximum clique of a bipartite graph is 2. A bipartite analogous equivalent of cliques is a **biclique**. A **biclique** of the graph G is a complete bipartite subgraph of G. [3]

1.2 Known results in Graph Decompositions

Graph decomposition has been a prominent research area in graph theory and combinatorics since the 1960s [22]. Although not referred to as a graph decomposition, graph decomposition and factorization can be seen in various combinatorial problems in the 19th century such as "Kirkman's 15 strolling school girls" [22], *Dudney's handcuffed prisoners* [22] and Euler's 36 army officer problem [22]. In 1966, Erdös, Goodman and Posa first introduced the concept of H-decomposition in their paper "The representation of a graph by set intersection" [19, 36]. The interest in graph decomposition is not surprising as graph decomposition has many real world application such as bioinformatics [30, 4], social science research, network and topology research [15], coding theory [14], and in many other computer science applications [6].

1.2.1 Graph Decomposition is NP-Complete

Given graphs G and H we may ask whether G decomposes into H. We call this the "Graph decomposition problem". According to Lonc [29], Ian Holyer in his dissertation "The computational complexity of Graph Theory problems" [24] conjectured in 1980 that the graph decomposition problem is NP-complete if the graph H has at least three edges. Holyer proved the conjecture for the cases where H is a complete graph and G is a simple circuit. Daniel Leven presented an unpublished proof for the case where H is a star. In 1991, Cohen and Tarsi extended the proof to include trees [12]. Finally in 1992 and 1995, Dor and Tarsi generalized the proof to include graphs that contains a connected component of at least three edges [16]. However, Holyer's conjecture was proven false when H is not a connected graph [17]. Bialoski and Roddity showed that the problem is polynomial when H is a set of three disjoint edges ($3K_2$ [see definition of a complete graph]). This was further generalized by Alon [1] where H is a set of s disjoint edges (sK_2). Favaron, Lonc and Truszczynski [20], also showed that the problem has polynomial complexity for the case where $H = K_{1,2} \bigcup K_2$ [17] . This result was further extended when Priesler and Tarsi [31] showed that the problem is still polynomial when H=$K_{1,2} \bigcup tK_2$.

The result of these findings gave strong evidence for a revised version of the Holyer's conjecture, that is, a H-decomposition of graphs is NP-complete if and only if the graph H contains a connected component of at least three edges [17].

1.2.2 Graph Decomposition of Complete Graphs

While the graph decomposition problem in general is NP-complete, by imposing conditions on the graphs G and H, researchers have proven the existence of certain H-decomposition should these criteria be met on the graph G. We first briefly give a survey of decomposition results into stars. In 1974, Cain showed that complete graphs K_n and K_{n+1} decompose into m-stars, if and only if m is odd or n is an even multiple of m and $n > m$ [8].

In 1974, Yamamoto, Ikeda, Shige-eda, Ushio and Hamada [38] showed that $K_{m,n}$ decomposes into k-stars if and only if k divides $m \times n$ for $k \leq m$, $k \leq n$ or k divides m or n . This result is later extended by Ushio and Yamamoto [37], who showed that there is a k-star decomposition for complete equal sized m-partite graphs of size n if $\frac{m \times (m-1)}{2} n^2$ divides c and $mn \geq 2c$. This result is then further extended by Shyu, [33] showing that a crown graph S_n^0 can be decomposed into $K_{l,m}$ if there is a positive integer value for λ such that $n = \lambda lm + 1$ [33]. In 2013, Lee and Lin [28] showed that a (C_k, S_K)-decomposition of crown graphs such that there is at least one copy of C_k and one copy of S_k when $4 \leq k \leq \frac{n-1}{2}$, k is even and k divides $n(n-1)$.

There has been some research into regular bipartite graphs, namely by Jacobson, Truszczynski and Tuza [26] who proved that a $2r$-regular bipartite graph has a decomposition into trees of size r. They also prove that every r-regular bipartite graph can be decomposed into double stars (a tree with 2 internal vertices and r leaves) of size r. They also proved that 4-regular bipartite graphs can be decomposed into paths of length 4. Moreover, they also proved that a $r-$dimensional cube decomposes into a tree of size r.

There has also been substantial research into the decomposition of complete bipartite graphs. In 1981, Sotteau [35] showed that there is a $2k$-cycle decomposition for all complete $K_{m,n}$ bipartite graphs if $2k$ divides mn, and both m and n are even, and $k < m$ and $k < n$. An extension of this result presented by Cichacz, Froncek, Kovar shows that a $K_{n,n}$ bipartite graphs can be decomposed into prisms [11].

There are many more proven decomposition for complete graphs such as decomposition into trees, (Lonc (1988), Yu Min Li (1990)), cycles (Farrell (1982)) and paths, however these decompositions are beyond the scope of this thesis. Further results on graph decompositions may be found in VI. 24 of Handbook of Combinatorial Designs [13].

1.2.3 Probabilistic Methods

As the problem of graph decomposition is conjectured to be NP-complete, especially when weak conditions are imposed on the graph G, we also look into the probabilistic method pioneered by Erdös in his paper "Graph Theory and Probability" published in 1959 [18] and expanded upon in 1961. Despite the name and the use of probability, the probabilistic method gives a conclusive proof on the existence (or the non-existence) of a mathematical object.

In their book "Probabilistic Method", Alon and Spencer state that the idea behind the probabilistic method is to create an appropriate probability space, and then show that a randomly chosen object has a positive probability to have specified properties in order to prove the existence of such object [2].

In the paper by Yuster [39], this method was used to show that there is H-decomposition where H is a tree with at least h vertices if the minimum degree of the graph $\delta(g)$ is greater than $\frac{|V|}{2} + 10h^4\sqrt{|V|\log|V|}$. It was shown that with the minimum degree, and by applying the Chernoff bound, there is a positive probability that the graph would have the required properties for such an H-decomposition.

We will explore whether Yuster's result can be strengthened in the case when H is a star and G is a bipartite graph in Section 4.2.

1.2.4 Solutions and Algorithms for S_1-decomposition and S_2-decomposition

Finding a H-Decomposition where of $H = S_1$ (equivalently $K_{1,1}$ or P_2) is trivial. Since there is only a single edge in the graph H, the set of edges $E(G)$ is itself the graph decomposition.

In the case of $H = S_2$ (equivalently $K_{1,2}$ or P_3), we first check if two divides $|E(G)|$ in each connected component. Having an even number of edges in each connected component is in fact the only necessary and sufficient criteria for a S_2 decomposition. First, we randomly assign directions to each of the edges and assign weight to each of the vertices in the graph by counting the number of directed edges pointing towards the vertex. Next, we find a pair of vertices with odd weights, and flip the direction of the edges in a path between these two vertices. Note that flipping the edges along the path does not change the parity of the weights of the vertices along the path, while changing the parity of the weights of the end vertices. We repeat this for every pair of vertices of odd weight. Finally we pair off the edges according to the direction of the edges to form copies of S_2, on the vertices with weight two and higher. This algorithm is folklore. Figure 1.16 illustrates this algorithm on a graph G, with the coloured lines representing the S_2 decomposition.

1.3 Representation of a decomposition in the thesis

In this section, we will explain how a graph decomposition is represented pictorially throughout the thesis.

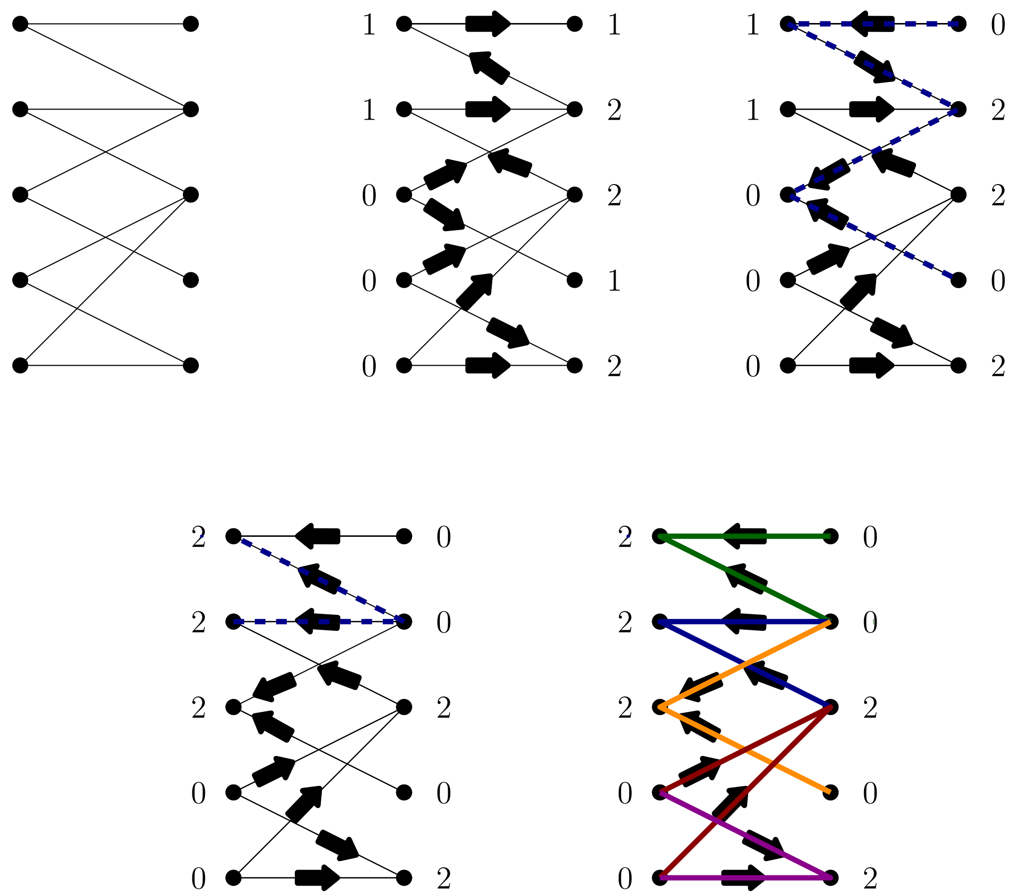


Figure 1.16: Polynomial time algorithm for S_2 decomposition.

Let U and V be 2 partite sets from a bipartite graph. In the illustration provided in figures 1.17, 1.18 the rows represents the vertices from partite set U and the columns represents the vertices from partite set V. A shaded area (possibly non-contiguous) of the same colour within a row or column of size r units, represents a copy of S_r.

In the cases where the bipartite graph is not complete, we denote the edges that are not part of the graph with a solid black region. In the cases where the graph has more than two partite sets, we will indicate the partite set in which the rows are represented on the left of the graph, and the partite set in which the columns are represented on the top of the graph.

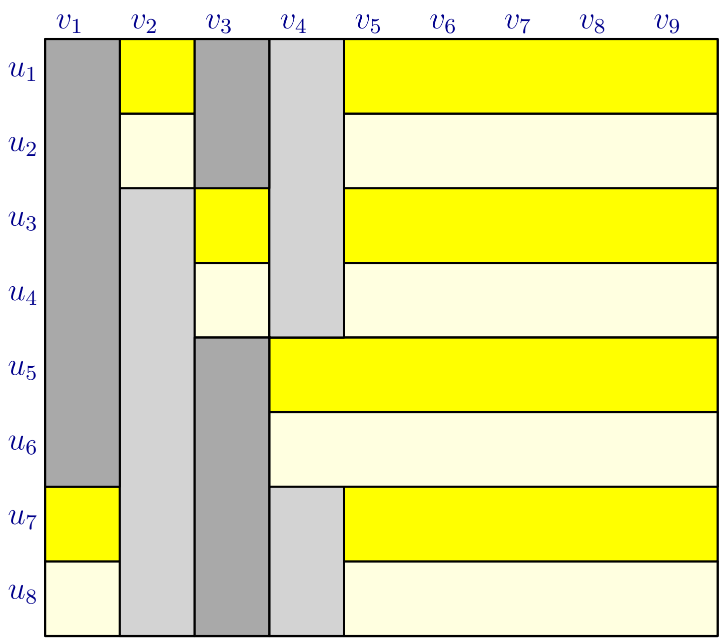


Figure 1.17: Graphical representation of the decomposition of the edges between partite set U and V

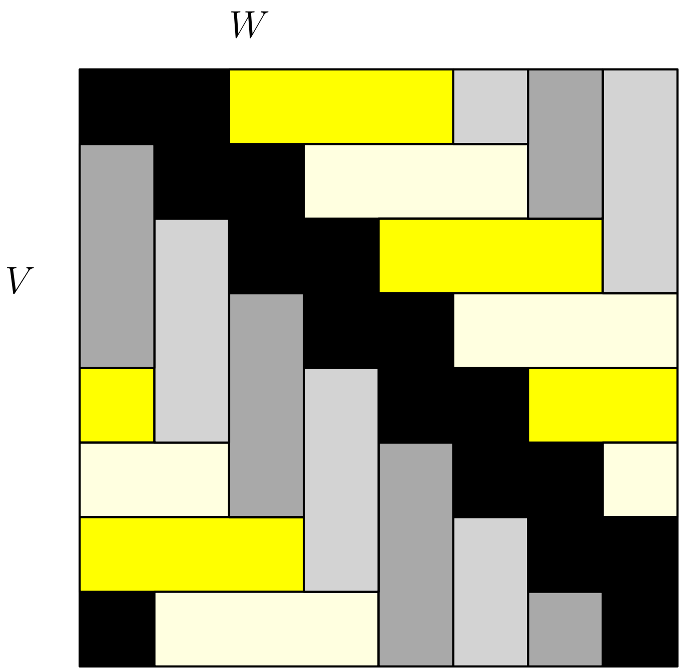


Figure 1.18: Graphical representation of the decomposition of the edges between partite set V and W when there are more than 2 partite sets and the graph is not complete

Chapter 2

Decomposition of complete Bipartite Graphs

In this section we give the necessary and sufficient conditions to decompose complete bipartite graphs and crown graphs into stars. Our proofs are by direct construction.

2.1 Preliminary Lemmas

Here we introduce some lemmas that will be used for S_r-decompositions of bipartite and multi-partite graphs.

Lemma 2.1 *If the degree of every vertex in a partite set U of a bipartite graph G is divisible by r, then there exists an S_r-decomposition of G.*

Proof. We can greedily choose r edges adjacent to a vertex in the partite set U to form a copy of S_r. We repeat this process until all the edges adjacent to the vertex are chosen. Then we repeat this process for each vertex in the partite set U until all the remaining edges have been chosen. $\square$

The following proof is an extension to Corollary 2.2 and 2.5 [9] that shows that if the graph $K_{m,m}$ decomposes into k-cycles, then the graph $K_{ml,ml} = K_{m,m} \otimes \overline{K_l}$ also decomposes into k-cycles. Moreover, if the graph $K_{m,m}$ decomposes into $k-$cycles, the graph $K_{ml,ml}$ also decomposes into kl-cycles.

Lemma 2.2 *If the graph G decomposes into S_r, there exists an S_r and an S_{rl} decomposition for the lexicographical product $G \otimes \overline{K_l}$.*

Proof. We let $H = S_r \otimes \overline{K_l}$. We then label the leaf vertices of S_r with integers from 1 to r, the center vertex of S_r as u and the vertices of $\overline{K_l}$ with integers from 1 to l. The resulting graph $H = S_r \otimes \overline{K_l}$ has the partite sets $U = \{u_y : 1 \leq y \leq l\}$, $V = \{v_{x,y} : 1 \leq x \leq r, 1 \leq y \leq l\}$ and edge set, $E(H) = \{e_{x,y,z} : 1 \leq x \leq r, 1 \leq y \leq l, 1 \leq z \leq l\}$ where $e_{x,y,z}$ is the edge between $v_{x,y}$ and u_z. Observe that H is isomorphic to $K_{l,rl}$.

Observe that the each vertex in the partite set U has degree rl. By Lemma 2.1, we can decompose H into S_r. Moreover, we can also decompose H into S_{rl}.

Formally, we partition the edges of H into graphs $H_{x,y}$ where $1 \leq x \leq l, 1 \leq y \leq l$ and

$$E(H_{x,y}) = \{e_{1,x,y}, e_{2,x,y}, \ldots, e_{r,x,y}\},$$

$$V(H_{x,y}) = \{v_{1,x}, v_{2,x}, \ldots, v_{r,x}, u_y\}.$$

Note that each $H_{x,y}$ is isomorphic to S_r. We can also partition the edges of H into graphs J_y where $1 \leq y \leq l$ and

$$E(J_y) = \{e_{1,1,y}, e_{2,1,y}, \ldots, e_{r,j,y}\},$$

$$V(J_y) = \{v_{1,1}, v_{2,1}, \ldots, v_{r,j}, u_y\}$$

and we also note that J_y is isomorphic to S_{rl}. □

2.2 Decomposition of Complete Square Bipartite Graphs

In this section we will prove that the complete bipartite graph $K_{p,p}$ has an S_r-decomposition if p^2 is divisible by r and r is less or equal to p by giving a construction of such decomposition. This theorem is also proven by Yamamoto, Ikeda, Shige-eda, Ushio and Hamada [38]. In the proof given in that paper, the authors showed that the bipartite graph $K_{m,n}$ can be represented as mn lattice points. From there, they showed that they can represent the decomposition using claw-type subsets of size r. They then show that each subset represents a claw or a S_r graph, and showed that there is always an arrangement for the subsets when the conditions above are met.

The construction of our proof here, although similar to the techniques given in the paper, was developed independently of the paper and is original.

Theorem 2.3 *The graph $K_{p,p}$ decomposes into S_r if and only if p^2 is divisible by r and $r \leq p$.*

Proof. We first show the necessity of the conditions $r \mid p^2$ and $r \leq p$. Suppose that r does not divide p^2. The number of edges in a $K_{p,p}$ graph is equal to the product of the number of vertices in the two partite set, i.e. p^2. By the definition of a decomposition, the number of edges of a decomposition of S_r must divide the number of edges in $K_{p,p}$ and therefore $r \mid p^2$.

Suppose $r > p$. We will show that $K_{p,p}$ has no subgraph isomorphic to S_r. Thus $K_{p,p}$ has no decomposition into S_r. Each vertex in $K_{p,p}$ has degree p. Therefore, any subgraph of $K_{p,p}$ has degree of at most p. Since S_r has a vertex degree of r, $K_{p,p}$ has no subgraph isomorphic to S_r.

We now show the conditions $r \mid p^2$ and $r \leq p$ are sufficient. We methodically divide the proof to according to the following cases:

Case 2.3.1: $r \mid p$.

Case 2.3.2: $r \nmid p$ and r is square.

Case 2.3.3: $r \nmid p$ and r is not square.

Case 2.3.1 *r divides p.*

Let $m = \frac{p}{r}$. Note that every vertex in the partite set V has degree mr. By Lemma 2.1, there is a S_r-decomposition of the graph.

Formally, let $U = \{u_{i,j} : 1 \leq i \leq m, 1 \leq j \leq r\}$, and let $V = \{v_k : 1 \leq k \leq p\}$ be the partite sets of $K_{p,p}$.

We can then define the S_r-decomposition of $K_{p,p}$ as follows:

$$V(H_{i,k}) = \{v_k, u_{i,j} : 1 \leq j \leq r\}$$

with $1 \leq i \leq m$ and $1 \leq k \leq p$.

Observe that each $H_{i,k}$ is isomorphic to S_r.

Case 2.3.2 *r does not divide p and r is square.*

Let $r = i^2$. Let $n = \frac{p-p'}{r}$ where $r \leq p' \leq 2r$, and let U and V be the two partite sets of $K_{p,p}$. We partition U into disjoint subsets U' and U'' such that $|U'| = p'$ and $|U''| = nr$. Similarly, we partition V into disjoint subsets V' and V'' such that $|V'| = p'$ and $|V''| = nr$.

By Lemma 2.1 we can partition the edges between U'' and V into copies of S_r. Similarly, by Lemma 2.1 we can partition the edges between V'' and U into copies of S_r. The remaining edges not partitioned by the steps above are the edges between U' and V'.

Since $r \mid p^2$, we have

$$r \mid (p' + nr)^2$$

$$\Rightarrow r \mid p'^2 + 2nrp' + 4n^2r^2$$

$$\Rightarrow r \mid p'^2$$

$$\Rightarrow i^2 \mid p'^2$$

$$\Rightarrow i \mid p'.$$

We let $j = \frac{p'}{i}$. Observe that, since $r \leq p' \leq 2r$, we have $i \leq j' \leq 2i$. We now let $b = j' - i$. Note that $0 \leq j'b \leq p'$. The proof for this is as follows.

$$i \leq j' \leq 2i$$

$$\Rightarrow 0 \leq j' - i \leq i$$

$$\Rightarrow 0 \leq j'(j' - i) \leq ij' = p'.$$

We also note that $ib = p' - r$. We partition U' into disjoint subsets $U_0, U_1, U_2 \ldots U_{j'-1}$ such that $|U_x| = i$ where $0 \leq x \leq j' - 1$. Since $0 \leq j'b \leq p'$, we can partition V' into disjoint subsets $V_0, V_1, V_2, \ldots V_{j'-1}$ and V_* such that $|V_x| = b$ where $0 \leq x \leq j'-1$ and $|V_*| = p' - jb$.

By Lemma 2.1, we can decompose the edges between $U_0, U_1 \ldots U_{i-1}$ and V_0 into copies of S_{i^2} with the vertices of V_0 as centers. We then repeat this for the edges between $U_x, U_x + 1 \ldots U_{x+i-1(\mod j')}$ and V_x, for $0 \leq x \leq j'-1$. We have used $bj'i^2$ edges altogether using vertices from V' regularly. Thus we have used $\frac{bj'i^2}{j'i} = ib$ edges incident with each vertex from U'. By Lemma 2.1, we can decompose the remaining edges using p' copies of S_r with each vertex in U' the center of one S_r.

Formally, let $U' = \{U_g : 0 \leq g \leq j'-1,\}$ where $U_g = \{u_{g,h} : 1 \leq h \leq i\}$. Let $V' = \{V_g, V^* : 0 \leq g \leq j'-1\}$, where $V_g = \{v_{g,h} : 1 \leq h \leq b\}$ and $V^* = \{x_l : 1 \leq l \leq p - j'b\}$. We can then define the decomposition as

$$V(H_{g,h}) = \{v_{g,h}\} \bigcup_{g \leq l \leq g+i} U_{l \bmod j'} \text{ where } v_{g,h} \in V_g$$

with the vertex $v_{g,h}$ the center of a copy of S_{i^2} and

$$V(H'_{g,h}) = \{u_{g,h}\} \bigcup_{g-i \leq l \leq g-1} V_{l \bmod j'} \cup V_* \text{ where } u_{g,h} \in U_g$$

with vertex $u_{g,h}$ the center of a copy of S_{i^2}.

We illustrate this in Figures 2.1, 2.2.

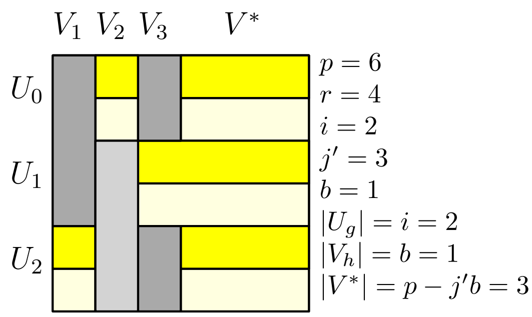


Figure 2.1: $K_{6,6}$ decomposes into S_4.

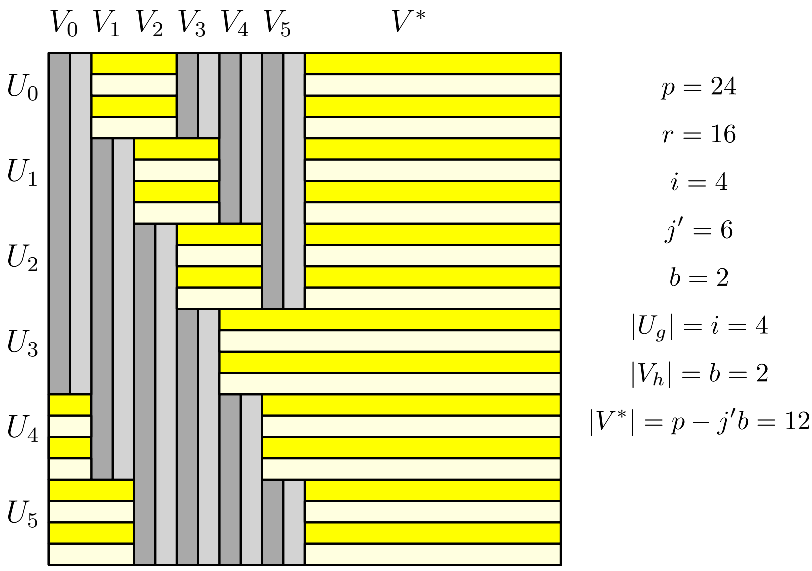


Figure 2.2: $K_{24,24}$ decomposes into S_{16}.

Case 2.3.3 *r does not divide p and r is not square.*

Let $r = i^2 j$ where j is a square free number,

$$r \mid p^2$$

$$\Rightarrow i^2 j \mid p^2$$

$$\Rightarrow ij \mid p.$$

We let $p = kij$.

We first observe that $K_{p,p}$ is the lexicographic product $K_{ik,ik} \otimes \overline{K_j}$. Since $r \leq p$, we have $i \leq k$. From Case 2.3.2, we have shown that $K_{ik,ik}$ decomposes into S_{i^2}. Using Lemma 2.2, it then follows that $K_{p,p}$ decomposes into S_{ji^2}. This is illustrated in Figure 2.3.

□

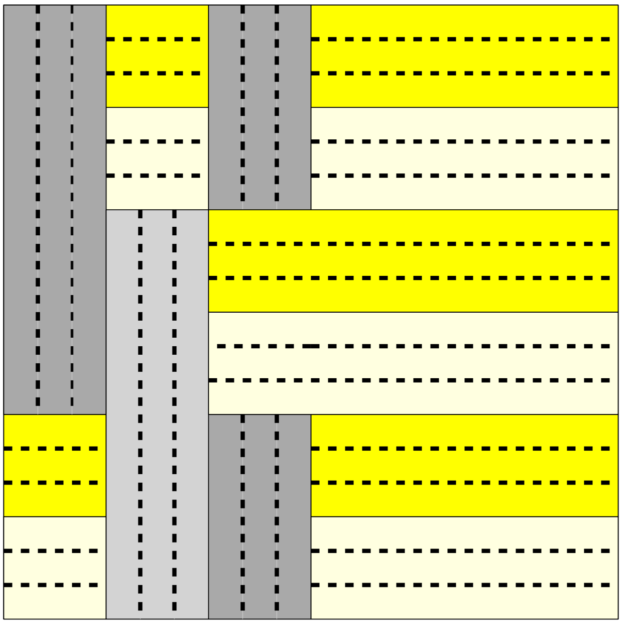


Figure 2.3: $K_{18,18}$ as the lexicographical product of $K_{4,4} \otimes \overline{K_3}$ decomposing into $S_4 \otimes \overline{K_3}$ and into S_{12}

2.3 Decomposition of Complete Bipartite Graphs

In this section we will show that the complete bipartite graph $K_{p,q}$ has a S_r-decomposition if at least one of the following two cases is satisfied:

Case 1: pq is divisible by r and $r \leq p$ and $r \leq q$.

Case 2: p is divisible by r or q is divisible by r.

As mentioned in the earlier section, this theorem was proven by Yamamoto, Ikeda, Shige-eda, Ushio and Hamada [38]. The construction of our proof here although similar to the techniques given in that paper, was developed independently of the paper and is original.

Theorem 2.4 *The complete bipartite graph $K_{p,q}$ decomposes into S_r if and only if one of the following cases is true:*

Case 1: pq is divisible by r and $r \leq p$ and $r \leq q$.

Case 2: p is divisible by r or q is divisible by r.

Proof. We first show the necessity of the conditions $r \mid pq$. Suppose that r does not divide pq. The number of edges in $K_{p,q}$ is equal to the product of the number of vertices in the two partite set, pq. By the definition of decomposition the number of edges in the decomposition must divide the number of edges in the graph; thus r must divide pq.

Now we will show the necessity of the condition $r \leq p$ and $r \leq q$ when $r \nmid p$ and $r \nmid q$. Without loss of generality let $p \geq q$, otherwise we swap the partite sets. Suppose $r \mid pq$, $r \nmid p$, $r \nmid q$ and $r > q$. Let c be the center vertex of a subgraph. Since the degree of each vertex in U is q and the degree of c is greater than q, c cannot be in U. However, since $r \nmid p$, there will be edges left over incident to vertex in V if all the center vertices in the S_r-decomposition belong to V. Therefore if $r > p$ or $r > q$ there is no S_r decomposition of $K_{p,q}$ in the case where $r \nmid p$ and $r \nmid q$.

From here, we can separate the proof to the following cases,

Case 2.4.1: r divides p or r divides q.

Case 2.4.2: r does not divide p and r does not divide q.

Case 2.4.1 *r divides p or r divides q.*

Without loss of generality, let r divide p, otherwise we swap the partite sets U and V.

Let $m = \frac{p}{r}$. Note that the vertex degree on every vertex of the partite set V is mr, therefore by Lemma 2.1, there is an S_r decomposition of the graph.

Case 2.4.2 *r does not divide p and r does not divide q.*

We let gcd(r,p)=i. This gives us, $r = ij$ and $p = ix$, gcd(j,x)= 1. Now,

$$r \mid pq$$

$$\Rightarrow ij \mid ixq$$

$$\Rightarrow j \mid xq$$

$$\Rightarrow j \mid q \text{ since } \gcd(j, x) = 1.$$

Therefore we have $i \mid p$ and $j \mid q$.

Let U and V be the partite sets of $K_{p,q}$ with U be size p and V size q respectively. Let $p' = p - nr$ where $r < p' < 2r$ and let $q' = q - mr$ where $r < q' < 2r$. We can partition U into two disjoint subsets U' and U'' such that $|U'| = p'$ and $|U''| = nr$. Similarly, we can partition V into two disjoint subsets V' and V'' such that $|V'| = q'$ and $|V''| = mr$.

By Lemma 2.1 we can partition the edges between U'' and V into copies of S_r. Likewise, by Lemma 2.1 we can partition the edges between V'' and U into copies of S_r.

Since $r \mid p^2$, we have

$$r \mid (p' + nr)(q' + mr)$$

$$\Rightarrow r \mid p'q' + nrq' + mrp' + mnr^2$$

$$\Rightarrow r \mid p'q'$$

$$\Rightarrow ij \mid p'q'.$$

We also have

$$i \mid p$$

$$\Rightarrow i \mid (p' + nr)$$

$$\Rightarrow i \mid p'$$

and

$$j \mid q$$

$$\Rightarrow j \mid (q' + mr)$$

$$\Rightarrow j \mid q'.$$

We let $k' = \frac{p'}{i}$ and $l' = \frac{q'}{j}$. Observe that $i < l' < 2i$. We let $b = l' - i$. Note that $0 < k'b \leq q'$ and the proof of this is as follows:

$$k'(l' - i) = k'l' - p' \leq q'$$

$$\iff k'l' \leq p' + q'$$

$$\iff \frac{p'q'}{r} \leq p' + q'.$$

We separate the remainder of the proof into two cases, $p \geq q$ and $p < q$:

Case i: $p' \geq q'$

$$\frac{p'q'}{r} < \frac{2rq'}{r} = 2q' \leq p' + q'$$
$$\iff q' \leq p'.$$

Case ii: $p' < q'$

$$\frac{p'q'}{r} < \frac{2rp'}{r} = 2p' \leq p' + q'$$
$$\iff p' \leq q'.$$

We also note that $jb = q' - r$.

We partition U' into disjoint subsets $U_0, U_1, U_2 \dots U_{k'-1}$ such that $|U_x| = i$ where $0 \leq x \leq k' - 1$. Since $0 < k'b \leq q'$, we can also partition V' into disjoint subsets $V_0, V_1, V_2, \dots V_{k'-1}$ and V^* such that $|V_x| = b$ where $0 \leq x \leq k' - 1$ and $|V^*| = q' - k'b$.

By Lemma 2.1, we can decompose the edges between $U_0, U_1 \dots U_{j-1}$ and V_0 into copies of S_{ij} with each vertex of V_0 a center of S_{ij}. We then repeat this for the edges between $U_x, U_{x+1} \dots U_{x+j-1(\mod k')}$ and V_x, for $0 \leq x \leq k' - 1$. We have used $ijbk'$ edges altogether using vertices from V' regularly. Thus we have used $\frac{ijbk'}{ik'} = jb$ edges incident with each vertex from U'. Our decomposition thus removes exactly $q' - r$ edges incident to each vertices in U'. By Lemma 2.1, we can decompose the remaining edges using p' copies of S_r with each vertex in U' the center of one S_r.

Figures 2.4 and 2.5, illustrates an example of this algorithm.

□

2.4 Decomposition of Crown Graphs

In this section, we extend the results of Theorem 2.3 and Theorem 2.4 to crown graphs. Here we show that a crown graph has a S_r-decomposition if and only if r divides $p^2 - p$ and r is less or equal to $p - 1$.

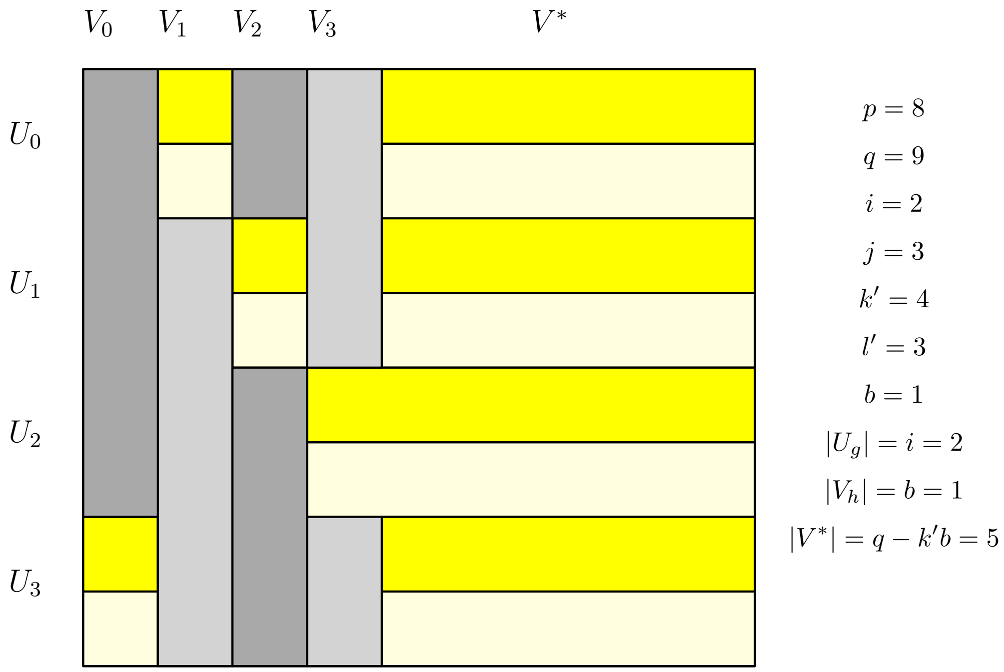


Figure 2.4: $K_{8,9}$ decomposing into S_6.

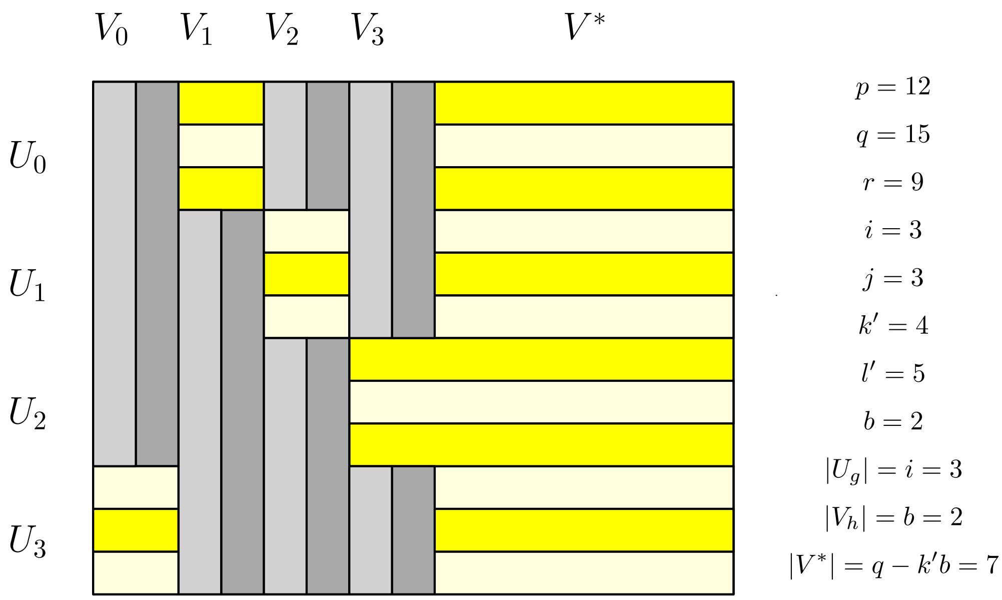


Figure 2.5: $K_{12,15}$ decomposing into S_9.

Theorem 2.5 *The crown graph $K_{p,p}$ minus a 1-factor decomposes into S_r if and only if $p^2 - p$ is divisible by r and $r \leq p - 1$.*

Proof. Observe that $K_{p,p}$ minus a 1-factor is isomorphic to S_p^0 (see Introduction).

We first show the necessity of the conditions $r \mid (p^2 - p)$ and $r \leq (p - 1)$. Suppose that r does not divide $p^2 - p$. The number of edges in S_p^0 equals $p^2 - p$.

By the definition of a decomposition, r must divide $p^2 - p$.

Suppose $r > p - 1$. We will show that S_p^0 has no subgraph isomorphic to S_r. Every vertex in S_p^0 has the degree $p-1$. Thus, every vertex in a subgraph of S_p^0 has degree at most $p-1$. Since the center vertex of S_r has a degree of r, S_p^0 has no subgraphs isomorphic to S_r.

From here, we can separate the proof to the following cases:

Case 2.5.1: r divides $p-1$.

Case 2.5.2: r divides p.

Case 2.5.3: r does not divide p, r does not divide $p-1$.

Case 2.5.1 *r divides $p-1$.*

Observe that each vertex in S_p^0 has the degree $p-1$. By Lemma 2.1, we can use a greedy algorithm to pick out the edges from one bipartite set to form $\frac{p^2-p}{r}$ copies of S_r.

Case 2.5.2 *r divides p.*

Let $m = \frac{p}{r}$ and let U and V be the 2 partite sets of the graph. We can partition V into m disjoint subsets $V_1, V_2, \ldots V_m$, each with size r. Let G_i be the subgraph induced by U and V_i where $1 \leq i \leq m$.

Observe that in each G_i, there are r vertices in partite set U with degree $r-1$ and $p-r$ vertices with degree r.

From here, we partition U into two disjoint subsets U_i' and U_i'' such that U_i'' is the set of $p-r-1$ vertices with degree r and U_i' is the set of r vertices with degree $r-1$ and one vertex with degree r. Let G_i'' be the subgraph induced by U_i'' and V_i. Observe that every vertex in U_i'' has degree r, and by Lemma 2.1 we can decompose the edges between U_i'' and V_i into S_r. We now define G_i' as the subgraph induced by U_i' and V_i. Observe again that each vertex in V_i in subgraph G_i' has degree r. By Lemma 2.1 we can decompose the edges of this subgraph into stars S_r. We repeat for each i, $1 \leq i \leq m$.

Figure 2.6 illustrates an example of this algorithm.

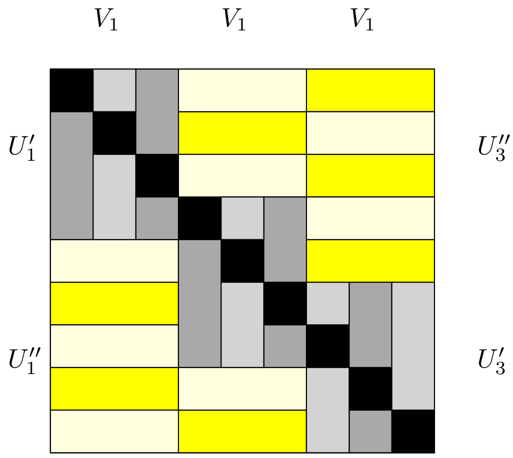


Figure 2.6: $K_{9,9}$ minus 1-factor decomposing into S_3.

Case 2.5.3 *r does not divide p and r does not divide $p-1$.*

Recall that $r \mid p(p-1)$. Let $\gcd(r,p) = i$, we then have

$$r = ij \text{ and } p = ix.$$

Now,

$$r \mid p(p-1)$$

$$\Rightarrow ij \mid ix(p-1)$$

$$\Rightarrow j \mid x(p-1)$$

since $\gcd(j,x) = 1$

$$\Rightarrow j \mid (p-1).$$

Let $n = \frac{p-p'}{r}$ where $r < p' < 2r$.

Since $n \geq 0$, we can partition the graph into a union of graphs $S^0_{p'} \cup nS^0_{r+1} \cup 2K_{p'-1,nr} \cup (n)(n-1)K_{r,r}$ as illustrated in Figure 2.7. By Lemma 2.1, we can decompose S^0_{r+1} (refer to case 2.5.1), and $K_{p'-1,nr}$ (refer to Theorem 2.4, case 2.4.1), $K_{r,r}$ (refer to Theorem 2.3, case 2.3.1) into S_r and the edges not partitioned are the edges in $S^0_{p'}$.

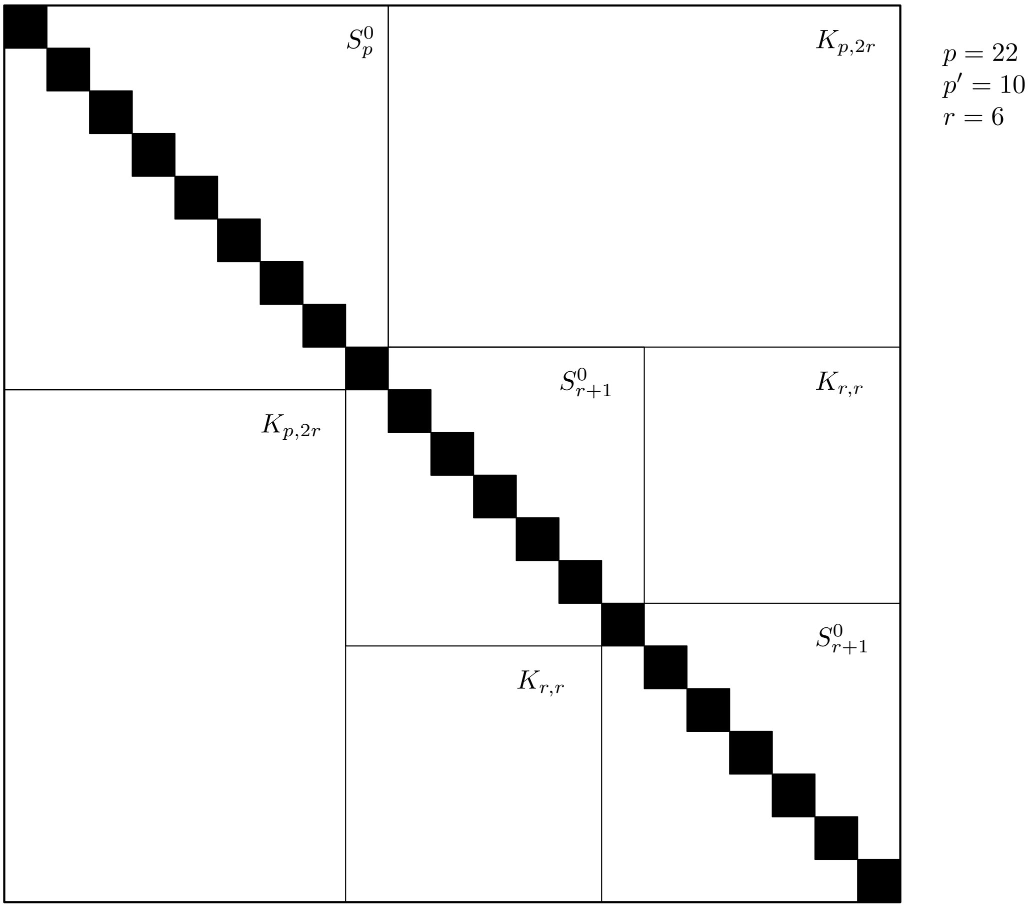


Figure 2.7: S_{22}^0 partitioned into subgraphs.

Observe that p' is divisible by i and $p' - 1$ is divisible by j. The proof of this is as follows. Since

$$i \mid p,$$

$$i \mid nr + p'$$

and since $r = ij$, we have

$$\Rightarrow i \mid p'.$$

Similarly,

$$j \mid (p - 1),$$

$$j \mid (nr + p' - 1)$$

and since $r = ij$, we have

$$\Rightarrow j \mid (p' - 1).$$

We let $x' = \frac{p'}{i}$ and $y' = \frac{p'-1}{j}$. Let $b = x'(y' - i)$.

Observe that $y' - i \geq 0$, since

$$r < p' < 2r$$

$$\Rightarrow r \leq p' - 1 < 2r$$

$$\Rightarrow ij \leq jy'$$

$$\Rightarrow j(y' - i) \geq 0.$$

Also observe that $\frac{j}{b}x' = p' - 1 - r$ the proof of which is as follows:

$$\frac{bj}{x'} = (y' - i)j$$

$$= jy' - ij$$

$$= p' - 1 - r. \qquad (2.1)$$

Let U and V be the partite sets of $S^0_{p'}$. We partition U into two disjoint subsets U_1 and U_2 such that $|U_1| = b$ and $|U_2| = p - b$. We then partition V into i disjoint subsets V_k of size x' where $1 \leq k \leq i$. For each vertex in U_1, we pick out j edges in each V_k, offsetting by one each time until we are done with each vertex in U_1.

We have used ijb edges altogether using vertices from U_1 regularly. Thus we have used $\frac{ijb}{ix'} = \frac{jb}{x'}$ edges incident with each vertex from V. Thus, our decomposition removes exactly $(p' - 1) - r$ edges incident to each vertices in V. By Lemma 2.1, the remaining edges forms p copies S_r using each vertex in V as the center vertex for one copy of S_r.

Formally, we let $U = U_1 \cup U_2$ where $U_1 = \{u_k : 1 \leq k \leq b\}$ and $U_2 = \{u_i : b + 1 \leq i \leq p\}$. Let $V = \bigcup_{1 \leq k \leq i} V_k$, where $V_k = \{v_{k,l} : 1 \leq l \leq x\}$. Let there be an edge between u_m and $v_{k,l}$ unless $kx + l = m$.

For each $1 \leq m \leq b$ we define the decomposition H_m to be

$$V(H_m) = \{u_m, v_{k,(l \bmod x)} : 1 \leq k \leq i, m + 1 \leq l \leq m + j + 1\}.$$

By equation (2.1) we have $\frac{bj}{x} = p' - 1 - r$ edges used up for every vertex in V. Therefore, we have exactly r edges incident to the vertices in V. By Lemma 2.1 we have an S_r-decomposition.

Figure 2.8 illustrates an example of this algorithm. □

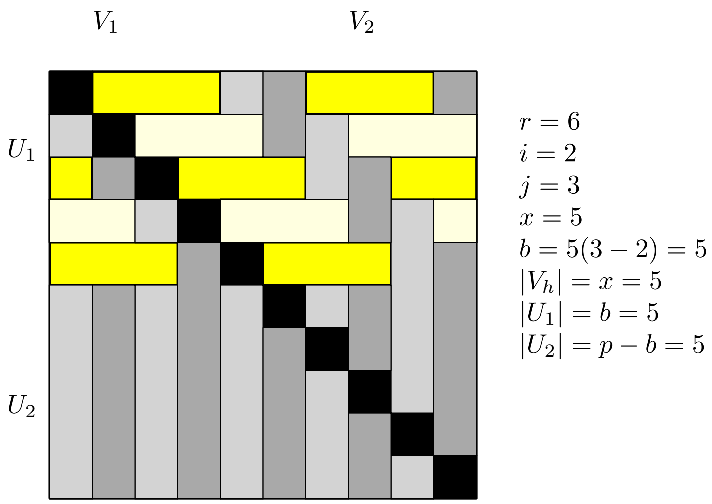


Figure 2.8: S^0_{10} decomposing into S_6.

Chapter 3

Decomposition of complete Tripartite Graphs

In this section, we give necessary and sufficient conditions to decompose complete equipartite tripartite graphs into stars. This result was proven by Ushio [37] in 1982. The proof by construction given below is original, and uses methods similar to those in Chapter 2. We will also extend the result for S_3-decompositions of $K_{p,q,r}$ where p, q and, r are not equal. We conclude this section by discussing how we might extend our results to S_r-decompositions of multipartite graphs.

3.1 Preliminary lemmas

Lemma 3.1 *If $\frac{a}{n} + \frac{b}{m} = 1$, there exists a decomposition of $K_{m,n}$ into m copies of S_a and n copies of S_b such that each vertex in the partite set of size m is the center of one copy of S_a and each vertex in the partite set of size n is the center of one copy of S_b.*

Proof. Without loss of generality, let $m \geq n$ otherwise we may swap the partite sets. To highlight the necessity of the condition, let $\frac{a}{n} + \frac{b}{m} = 1$; then multiplying mn to both sides gives us $ma+nb = mn$. Since the total number of edges of the m copies of S_a and n copies of S_b must equal the number of edges in $K_{m,n}$ this condition is necessary. We can then construct a decomposition to partition the edges into m copies of S_a and n copies of S_b. Let U and V be the two partite sets of $K_{m,n}$ containing m and n vertices respectively. Observe that vertices in U each have degree n and the vertices in V each have degree m. We use each vertex of U as the center vertex of a star S_a, offsetting each of the vertices used in V by one each time. This uses $a\frac{m}{n}$ edges incident with each of the n vertices of V. Since

$$\frac{a}{n} + \frac{b}{m} = 1$$

$$\Rightarrow a\frac{m}{n} = m - b,$$

there are exactly b edges incident with each of the vertices of V. By Lemma 2.1, we can then pick out the remaining b edges incident to each vertex of V creating n copies of S_b.

Formally, the decomposition is as follow. Let $U = \{u_i : 1 \leq i \leq m\}$ and $V = \{v_i : 1 \leq i \leq n\}$ then

$$V(H_i) = \{u_i, v_{c \bmod m} : i \leq c \leq (i + a - 1)\}$$

$$V(H'_j) = \{v_j, u_{c \bmod n} : j - b \leq c \leq j - 1\}$$

where $1 \leq i \leq m$ and $1 \leq j \leq n$. Observe that each H_i is isomorphic to S_a and each H'_j is isomorphic to S_b. □

Lemma 3.2 *If $K_{p,p,p}$ has a S_r-decomposition then at least p vertices are centers of S_r in two of the three partite sets.*

Proof. We let U, V, W be the 3 partite sets of $K_{p,p,p}$. We then define $c(X)$ to the be the number of vertices chosen to be a center of S_r in partite set X. Let $a = c(U)$, $b = c(V)$ and $c = c(W)$. Without loss of generality, let us assume that there exists an S_r-decomposition with $a < p$ and $b < p$. Since a and b are less than p, there exists vertices $u \in U$ and $v \in V$ that are not chosen to be centers of S_r. Note that every edge of S_r is an edge between the center and a leaf vertex. However, since both u and v are not the center vertex of some

S_r, the edge $\{u, v\}$ cannot be in a S_r decomposition. This is a contradiction, therefore both $a < p$ and $b < p$ cannot be true. □

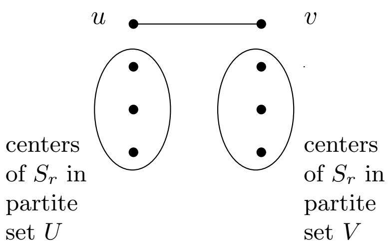


Figure 3.1: Vertex u and v not picked as centers

3.2 Decomposition of equipartite tripartite graphs

In this section we will prove that the complete tripartite graph $K_{p,p,p}$ has a S_r-decomposition if and only if $3p^2$ is divisible by r and r is less or equal to $\frac{2p}{3}$. We will provide a proof by construction of such decomposition. This theorem was also proven by Ushio, Tazawa, and Yamamoto [37]. In the proof by given in that paper, the authors showed that an adjacency matrix admits a S_r decomposition if the row sum vector equals r. The authors then showed that for all equipartite multipartite graphs, this condition is met when the necessity conditions are met.

Theorem 3.3 *The complete tripartite graph $K_{p,p,p}$ decomposes into S_r if and only if $3p^2$ is divisible by r and $p \geq \frac{2}{3}r$ and $r \mid 3p^2$.*

Proof. We first show the necessity of the conditions $r \mid 3p^2$ and $p \geq \frac{2}{3}r$. Let U, V, W be the three partite sets of $K_{p,p,p}$. Note that the graphs formed from the edges between U and V, V and W, and U and W, are each isomorphic to $K_{p,p}$. Hence the total number of edges in $K_{p,p,p}$ is $3p^2$. By the definition of decomposition, the edges in a decomposition must divide the total number of edges in the graph. Thus, r must divide $3p^2$.

Suppose $p < \frac{2}{3}r$. Let $c(X)$ be the number of vertices chosen to be a center of S_r in partite set X, and let $a = c(U)$, $b = c(V)$ and $c = c(W)$. By Lemma 3.2, at least two of the three partite sets have p vertices chosen as centers of S_r. Without loss of generality let $a \geq p$ and $b \geq p$. Also note that $r(a+b+c) = 3p^2$. We can then derive the following inequality:

$$r(a+b+c) = 3(p^2)$$

$$\Rightarrow r(p+p+c) \leq 3(p^2)$$

$$\Rightarrow r(2p+c) \leq 3(p^2).$$

Since it is impossible for c to be negative, we have the following;

$$r(2p) \leq 3(p^2)$$

$$\Rightarrow 2r \leq 3(p)$$

$$\Rightarrow p \geq \frac{2}{3}r.$$

We now show the sufficiency of the conditions, by separating proofs into the following cases:

Case 3.3.1: gcd$(r, 3)$= 3, $r = 3j, j \mid p$.

Case 3.3.2: gcd$(r, 3)$= 3, $r = 3k, k \nmid p$.

Case 3.3.3: gcd$(r, 3)$= 1, $r \leq p$.

Case 3.3.4: gcd$(r, 3)$= 1, $\frac{2}{3}r \leq p \leq r$.

Case 3.3.1 *gcd*$(r, 3)$*=* 3*,* $r = 3j, j \mid p$*.*

Let $n = \frac{p}{j}$. By Lemma 2.2, since $K_{p,p,p} = K_{n,n,n} \otimes \overline{K_j}$, if $K_{n,n,n}$ decomposes into S_3 then $K_{p,p,p}$ decomposes into S_{3j} for all $p \geq 2j$. Let U, V, W be the 3 partite sets of $K_{n,n,n}$.

There exists $a \geq 0$ and $0 \leq b \leq 2$ that satisfies $n = 3a + 2b$ for all $n \geq 2$, since $\gcd(3, 2) = 1$. We first pick out a total of b edge disjoint 1-factors between partite sets U and V, and a total of $2b$ edge disjoint 1-factors between partite sets U and W. Note that we can use these edges to form b copies of S_3 using each vertex in partite set U as a center. We then pick out another b edge-disjoint 1-factors between partite sets U and V, and $2b$ edge disjoint 1-factors between partite sets V and W. We also note that we can use these edges to form b copies of S_3 using each vertex in partite set V as a center. Observe each vertex in U is now incident with $3a$ edges between partite sets U and V. Also observe that each vertex in V is now incident with $3a$ edges between partite sets V and W, and each vertex in W is also incident with $3a$ edges between partite set W and U. By Lemma 1, we have a S_3 decomposition of the remaining edges.

Case 3.3.2 *gcd*$(r, 3)$= 3, $r = 3k, k \nmid p$.

Let $k = i^2 j$ where j is square-free. Since $r \mid 3p^2$,

$$k \mid p^2$$

$$\Rightarrow i^2 j \mid p^2$$

$$\Rightarrow ij \mid p$$

$$\Rightarrow p = nij.$$

Let $ni = \frac{p}{j}$. By Lemma 2.2,since $K_{p,p,p} = K_{ni,ni,ni} \otimes \overline{K_j}$, if $K_{ni,ni,ni}$ decomposes into S_{3i^2} then $K_{p,p,p}$ decomposes into S_r. Using the strategy from Case 3.3.1, we can divide the decomposition problem into partial decompositions of $3K_{ni,ni}$. By the necessary conditions, we have $ni \geq \frac{2}{3}(3i^2)$; we can then simplify this to $n \geq 2i$. We can now show a proof by construction of the existence of a S_{3i^2}-decomposition. Let us assume that there exists a S_{3i^2}-decomposition with a copies of S_{3i^2} with centers in partite set U each using x edges to V; b copies of S_{3i^2} with centers in partite set V each using y edges to W and c copies of S_{3i^2} with centers in partite set W each using z edges to U.

By summing the edges between partite sets U and V we have the following equality

$$a(x) + b(3i^2 - y) = (ni)^2. \tag{3.1}$$

By considering the edges between partite sets V and W we have

$$b(y) + c(3i^2 - z) = (ni)^2. \tag{3.2}$$

By considering the edges between partite sets U and W we have

$$c(z) + a(3i^2 - x) = (ni)^2. \tag{3.3}$$

Summing the three equations gives us

$$(3i^2)(a + b + c) = 3(ni)^2$$

$$\Rightarrow (3i^2)(a + b + c) = 3(ni)^2$$

$$\Rightarrow a + b + c = n^2. \tag{3.4}$$

The values of x,y and z are bound by the following

$$0 \leq x \leq \min(3i^2, ni);$$

$$0 \leq y \leq \min(3i^2, ni);$$

$$0 \leq z \leq \min(3i^2, ni).$$

Moreover we also have the following bounds

$$0 \leq 3i^2 - x \leq \min(3i^2, ni);$$

$$0 \leq 3i^2 - y \leq \min(3i^2, ni);$$

$$0 \leq 3i^2 - z \leq \min(3i^2, ni).$$

Now,

$$0 \geq x - 3i^2 \geq -\min(3i^2, ni)$$

$$\Rightarrow 3i^2 \geq x \geq \max(0, 3i^2 - ni).$$

We obtain similar bounds on y and z. Combining these bounds gives us

$$\max(3i^2 - ni, 0) \leq x \leq \min(3i^2, ni); \tag{3.5}$$

$$\max(3i^2 - ni, 0) \leq y \leq \min(3i^2, ni); \tag{3.6}$$

$$\max(3i^2 - ni, 0) \leq z \leq \min(3i^2, ni). \tag{3.7}$$

We then consider the following sub-cases:

Case: 3.3.2.1 $2i \leq n \leq 3i$.

Case: 3.3.2.2 $3i \leq n \leq 4i$.

Case: 3.3.2.3 $4i \leq n \leq 5i$.

Case: 3.3.2.4 $n \geq 5i$.

Case 3.3.2.1 $2i \leq n \leq 3i$.

Using inequalities (3.5), (3.6), (3.7) we have the following bounds for x,y,z for $2i \leq n \leq 3i$:

$$3i^2 - ni \leq x \leq ni;$$

$$3i^2 - ni \leq y \leq ni;$$

$$3i^2 - ni \leq z \leq ni. \tag{3.8}$$

We now set the following,

$$a = ni,$$

$$b = ni,$$

$$c = n^2 - 2ni,$$

$$x = n^2 - 3ni + 3i^2,$$

$$y = n^2 - 4ni + 6i^2,$$

$$z = ni.$$

We will now show that our choice above satisfies equations (3.1), (3.2), (3.3), (3.4), and the inequalities (3.8). Looking at equation 3.4, we have

$$a + b + c = ni + ni + (n^2 - 2ni) = n^2.$$

We can also show that equations (3.1), (3.2), (3.3) are satisfied by our choice of a, b, c, x, y, and z. The left hand side of equation (3.1) is equal to

$$a(x) + b(3i^2 - y)$$

$$= ni(n^2 - 3ni + 3i^2) + ni(3i^2 - (n^2 - 4ni + 6i^2))$$

$$= ni(n^2 - 3ni + 3i^2) + ni(4ni - 3i^2 - n^2)$$

$$= n^2 i^2.$$

Again, the left hand side of equation (3.2) is equal to

$$b(y) + c(3i^2 - z)$$

$$= ni(n^2 - 4ni + 6i^2) + (n^2 - 2ni)(3i^2 - (ni))$$

$$= ni(n^2 - 4ni + 6i^2) + ni(n - 2i)(3i - n)$$

$$= ni(n^2 - 4ni + 6i^2 - (n^2 - 5ni + 6i^2))$$

$$= n^2 i^2.$$

Finally, the left hand side of equation (3.3) is equal to

$$a(3i^2 - x) + c(z)$$

$$= ni(3i^2 - (n^2 - 3ni + 3i^2)) + (n^2 - 2ni)(ni)$$

$$= ni(3ni - n^2) + (n^2 - 2ni)(ni) = n^2 i^2.$$

Now we can show that our choice of x, y, z satisfies bounds given by inequalities (3.8) for $2i \leq n \leq 3i$. Checking for the lower bounds for x we have

$$x \geq 3i^2 - ni$$

$$\iff n^2 - 3ni + 3i^2 \geq 3i^2 - ni$$

$$\iff n^2 - 2ni \geq 0$$

$$\iff n(n - 2i) \geq 0$$

$$\iff n \leq 0 \text{ or } n \geq 2i.$$

Checking for the upper bounds for x we have

$$x \leq ni$$

$$\iff n^2 - 3ni + 3i^2 \leq ni$$

$$\iff n^2 - 4ni + 3i^2 \leq 0$$

$$\iff (n-3i)(n-i) \leq 0$$

$$\iff i \leq n \leq 3i \text{ which is true because } 2i \leq n \leq 3i.$$

Moreover, for $2i \leq n \leq 3i$, we can also show that the lower bound of y is $y \geq ni - c \geq r - ni$. We first show the second inequality

$$ni - c \geq r - ni$$

$$\iff ni - n(n-2i) \geq r - ni$$

$$\iff n(4i - n) - 3i^2 \geq 0$$

$$\iff n^2 - 4ni + 3i^2 \leq 0$$

$$\iff (n-3i)(n-i) \leq 0$$

$$\iff i \leq n \leq 3i.$$

Now we verify that $y \geq ni - c$

$$n^2 - 4ni + 6i^2 \geq ni - c$$

$$\iff n^2 - 4ni + 6i^2 \geq ni - n(n-2i)$$

$$\iff 2n^2 - 7ni + 6i^2 \geq 0$$

$$\iff 2n^2 - 7ni + 6i^2 \geq 0$$

$$\iff (2n-3)(n-2i) \geq 0$$

$$\iff n \leq \frac{3}{2}i \text{ or } n \geq 2i \tag{3.9}$$

which is true because $2i \leq n \leq 3i$.

Looking at the upper bounds of y, we have

$$n^2 - 4ni + 6i^2 \leq ni$$

$$\iff n^2 - 5ni + 6i^2 \leq 0$$

$$\iff n^2 - 5ni + 6i^2 \leq 0$$

$$\iff (n-2i)(n-3i) \leq 0$$

$$\iff 2i \leq n \leq 3i.$$

Finally $z = ni$ clearly satisfies the inequality $r - ni \leq z \leq ni$.

Observe that for $2i \leq n \leq 3i$,

$$0 \leq c \leq ni. \tag{3.10}$$

The proof of which is as follows:

$$n(n-2i) \geq 0$$

$$\Rightarrow n \leq 0 \text{ or } n \geq 2i;$$

$$n(n-2i) \leq ni$$

$$\iff n(n-3i) \leq 0$$

$$\Rightarrow 0 \leq n \leq 3i.$$

By equation (3.1), we have

$$a(x) - b(3i^2 - y) = n^2 i^2$$

$$\Rightarrow ni(x) - ni(3i^2 - y) = n^2 i^2.$$

Dividing both sides by $n^2 i^2$ gives us

$$\Rightarrow \frac{x}{ni} - \frac{3i^2 - y}{ni} = 1.$$

Thus by Lemma 3.1, the edges between U and V can be decomposed into ni copies of S_x and ni copies of S_{3i^2-y} so that each vertex of U is the center of one copy of S_x and each vertex of V is the center of one copy of S_{3i^2-y}.

Let D_{uv} be the set of S_x's and D_{vu} be the set of S_{3i^2-y}'s in this decomposition.

We next partition W into disjoint sets W' and W'', such that $|W'| = c$ and $|W''| = ni - c$. Observe that $3i^2 - x = ni - c$:

$$3i^2 - (n^2 - 3ni + 3i^2)$$

$$= 3ni - n^2$$

$$= ni + 2ni - n^2$$

$$= ni - (n^2 - 2ni)$$

$$= ni - c$$

By Lemma 2.1 we can decompose the edges between U and W'' into $a = ni$ copies of S_{3i^2-x} with each vertex of U the center of one copy of S_{3i^2-x}. By Lemma 2.1, we can also decompose the edges between U and W'' into c copies of $S_{z=ni}$ with each vertex of W'' the center of one copy of S_z. We let D_{uw} be the set of S_{3i^2-x}'s and D_{wu} be the set of S_z's in this decomposition.

Again, by Lemma 2.1, we can decompose the edges between V and W'' into ni copies of S_{ni-c} with each vertex of V the center of one copy of S_{ni-c}. We will now show that by Lemma 3.1 we have a decomposition between the edges of V and W' with ni copies of S_{y-ni+c} with each vertex of V the center of one copy of S_{y-ni+c} and c copies of S_{3i^2-z} with each vertex of W' the center of one copy of S_{3i^2-z}

$$\frac{y - ni + c}{c} + \frac{3i^2 - z}{ni}$$

$$= \frac{n^2 - 4ni + 6i^2 - ni}{n^2 - 2ni} + 1 + \frac{3i^2 - ni}{ni}$$

$$= \frac{n^2 - 5ni + 6i^2}{n^2 - 2ni} + \frac{3i - n}{n} + 1$$

$$= \frac{(n^2 - 5ni + 6i^2) + (n - 2i)(3i - n)}{n(n - 2i)} + 1$$

$$= \frac{(n - 2i)(n - 3i) + (n - 2i)(3i - n)}{(n^2 - 2ni)} + 1$$

$$= 1.$$

Let $D_{vw''}$ be the set of S_{ni-c}'s, $D_{vw'}$ be the set of S_{y-ni+c}'s and D_{wv} be the set of S_{3i^2-z}'s.

We now let $D_u = D_{uv} \cup D_{uw}$, observe that each vertex in U is the center of one copy of S_x and one copy of S_{3i^2-x}, the union of which is isomorphic to S_{3i^2}. Similarly, we let $D_v = D_{vu} \cup D_{vw'} \cup D_{vw''}$; each vertex in V is the center of one copy of S_{3i^2-y}, one copy of S_{ni-c} and one copy of S_{y-ni+c}, the union of which is S_{3i^2}. Finally, we let $D_w = D_{wu} \cup D_{wv}$, and note that each vertex in W' is the center of one copy of S_z, and one copy of S_{3i^2-z}, the union of which gives us S_{3i^2}.

Note that any positive integer solution for a, b, c, x, y, and z that satisfy equations (3.1), (3.2), (3.3), (3.4) while fulfilling the bounds given in 3.8 can construct a S_{3i^2} decomposition.

Case 3.3.2.2 : $3i \leq n \leq 4i$.

Let $q = n - 3i$ and $n' = n - 2q$. Observe that $0 \leq q \leq i$ and $2i \leq n' \leq 3i$ when $3i \leq n \leq 4i$.

Let U, V and W be the partite sets of $K_{ni,ni,ni}$. We partition U into three disjoint subsets U_1, U_2 and U_3; V into three disjoint subsets V_1, V_2 and V_3 and W into three disjoint subsets W_1, W_2 and W_3 such that $|U_1| = |U_2| = |V_1| = |V_2| = |W_1| = |W_2| = qi$ and $|U_3| = |V_3| = |W_3| = ni - 2qi$. Let $U' = U_1 \cup U_3$, $U'' = U_2 \cup U_3$, $V' = V_1 \cup V_3$, $V'' = V_2 \cup V_3$, $W' = W_1 \cup W_3$ and $W'' = W_2 \cup W_3$.

Observe that $n - q = 3i$ and $|U'| = |U''| = |V'| = |V''| = |W'| = |W''| = 3i^2$. By Lemma 2.1 we can decompose the edges between U_1 and V' using qi copies of S_{3i^2} with each vertex in U_1 the center of one copy of S_{3i^2}. Similarly, we can decompose the edges between U_2 and V'', using qi copies of S_{3i^2} with each vertex in U_2 the center of one copy of S_{3i^2}; the edges between V_1 and U''; with each vertex in V_1 the center of one copy of S_{3i^2} and the edges between V_2 and U' with each vertex in V_2 the center of one copy of S_{3i^2}, by Lemma 2.1. An example of this decomposition is illustrated in Figure 3.2.

We repeat this for each pair of partite sets. The remaining set of edges that is not decomposed in the steps above is isomorphic to $K_{n'i,n'i,n'i}$. We can then decompose this graph by referring to case 3.3.2.1.

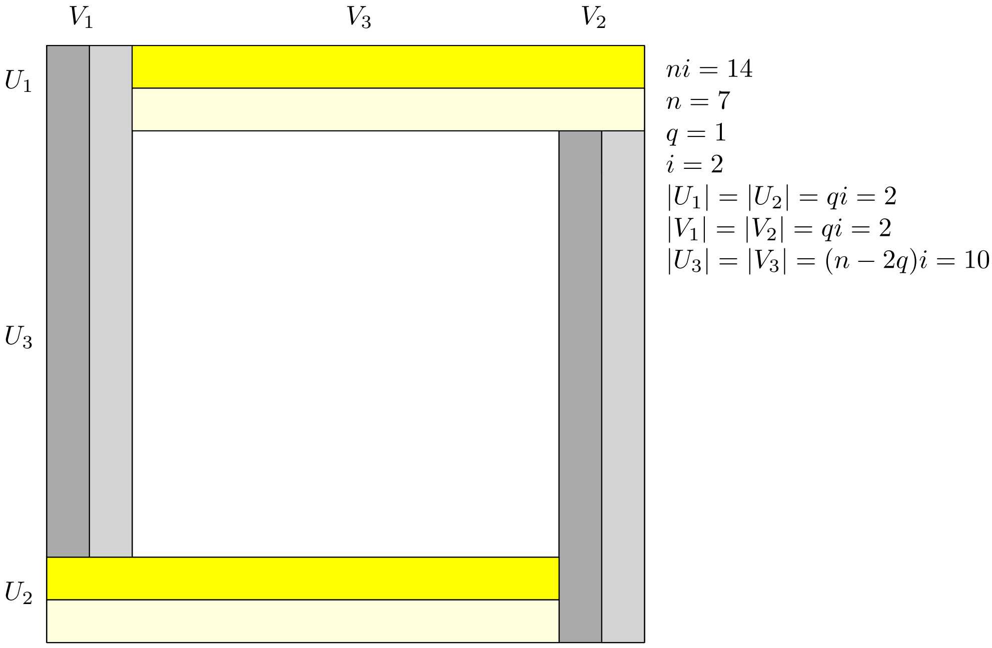


Figure 3.2: $K_{14,14}$ reduced to $K_{10,10}$.

Case 3.3.2.3 $4i \leq n \leq 5i$.

Initially, we planned to use the strategy from Case 3.3.2.2 to reduce the case into $n' = n - 2i$, however while constructing the decomposition, it became apparent that this strategy did not work for odd values of i. We can however construct a new proof by construction using the techniques from case 3.3.2.1.

Let us assume that there exists a S_{3i^2}-decomposition with $2ni$ copies of S_{3i^2} with centers in partite set U where each vertex is a center of two copies of S_{3i^2}, such that one copy has x_1 edges to V and the other copy has x_2 edges to V, and $2ni$ copies of S_{3i^2} with centers in partite set V, where each vertex is the center of two copies of S_{3i^2} such that one copy has y_1 edges the other copy has y_2 edges to W; and c copies of S_{3i^2} with c vertices of W a center of one copy of S_{3i^2} in partite set W with z edges to U.

For $4i \leq n \leq 5i$, the bounds given by inequalities (3.5,(3.6),(3.7) gives us

$$0 \leq x_1 \leq 3i^2;$$

$$0 \leq x_2 \leq 3i^2;$$

$$0 \leq y_1 \leq 3i^2;$$

$$0 \leq y_2 \leq 3i^2;$$

$$0 \leq z \leq 3i^2. \quad (3.11)$$

Moreover, by the decomposition described above, we have these additional bounds

$$6i^2 - ni \leq x_1 + x_2 \leq ni;$$

$$6i^2 - ni \leq y_1 + y_2 \leq ni. \quad (3.12)$$

We now set the following:

$$a = ni;$$

$$b = ni;$$

$$c = n^2 - 4ni;$$

$$x_1 = i^2;$$

$$x_2 = i^2;$$

$$y_1 = i^2;$$

$$y_2 = 7i^2 - ni;$$

$$z = i^2.$$

Looking at the edges between partite sets U and V and referring to equality (3.1), we have

$$ni(x_1) + ni(x_2) + ni(3i^2 - y_1) + ni(3i^2 - y_2)$$

$$= ni(i^2 + i^2 + (3i^2 - i^2) + (3i^2 - (7i^2 - ni)))$$

$$= ni(7i^2 + ni - 7i^2) = n^2i^2. \quad (3.13)$$

Looking at the edges between partite sets V and W and referring to equality (3.2), we have

$$ni(y_1) + ni(y_2) + (c)(3i^2 - z)$$

$$= ni(i^2 + (7i^2 - ni)) + (n^2 - 4ni)(2i^2)$$

$$= ni(8i^2 - ni) + ni((n - 4i)(2i))$$

$$= ni(8i^2 - ni + 2ni - 8i^2)$$

$$= n^2i^2.$$

Looking at the edges between partite sets W and U and referring to equality (3.3), we have

$$c(z) + ni(3i^2 - x_1) + ni(3i^2 - x_2)$$

$$= (n^2 - 4ni)(i^2) + ni(3i^2 - i^2) + ni(3i^2 - i^2)$$

$$= ni(ni - 4i^2) + ni(4i^2)$$

$$= n^2i^2.$$

Observe that $x_1 = x_2 = y_1 = z = i^2$ fulfils the bounds given in inequalities (3.11). Also observe that $y_2 = 7i^2 - ni$ fulfils the bound $0 \leq y_2 \leq 3i^2$ for $4i \leq n \leq 5i$.

We also observe that inequalities (3.12) are satisfied by our choice of x_1, x_2, y_1 and y_2.

$$6i^2 - ni \leq x_1 + x_2 \leq ni$$

$$\Longleftrightarrow \; 6i^2 - ni \leq 2i^2 \leq ni$$

$$\Longleftrightarrow \; n \geq 4i.$$

$$6i^2 - ni \leq y_1 + y_2 \leq ni$$

$$\Longleftrightarrow \; 6i^2 - ni \leq i^2 + 7i^2 - ni \leq ni$$

$$\Longleftrightarrow \; 8i^2 \leq 2ni$$

$$\Longleftrightarrow \; n \geq 4i.$$

Note that $y_1 + y_2 \geq ni - c$ for $ni \geq 4i$, the proof of which is as follows:

$$y_1 + y_2 \geq ni - c$$

$$\iff i^2 + 7i^2 - ni \geq ni - (n^2 - 4ni)$$

$$\iff 8i^2 - ni \geq 5ni - n^2$$

$$\iff n^2 - 6ni + 8i^2 \geq 0$$

$$\iff (n - 4i)(n - 2i) \geq 0$$

$$\iff n \geq 4i \text{ or } n \leq 2i$$

Observe that $x_1 + x_2 = 2i^2$ and $6i^2 - y_1 - y_2 = ni - 2i^2$. Dividing equation (3.13) by n^2i^2 gives us the necessity condition for Lemma 3.1:

$$\frac{x_1}{ni} + \frac{x_2}{ni} + \frac{3i^2 - y_1}{ni} + \frac{3i^2 - y_2}{ni}$$
$$= \frac{x_1 + x_2}{ni} + \frac{6i^2 - y_1 - y_2}{ni}$$
$$= 1.$$

By Lemma 3.1, there exists a decomposition of the edges between U and V using $a = ni$ copies of $S_{(x_1+x_2)=2i^2}$ with each vertex of U the center of a copy of S_{2i^2} and $b = ni$ copies of $S_{(6i^2-y_1-y_2)=ni-2i^2}$ with each vertex of V the center of a copy of S_{ni-2i^2}.

Let D_{uv} be the set of S_{2i^2}'s and D_{vu} be the set of S_{ni-2i^2}'s in this decomposition.

We can partition W into disjoint sets W' and W'', such that $|W'| = c = n^2 - 4ni$ and $|W''| = ni - c = 5ni - n^2$. By Lemma 2.1 we can decompose the edges between U and W'' into ni copies of S_{5ni-n^2} with each vertex of U the center of one copy of S_{5ni-n^2}.

Let $k = n^2 - 5ni + 4i^2$, observe that k is positive for all $n \geq 4i$. Also observe that $k + 5ni - n^2 = 4i^2 = 6i^2 - x_1 - x_2$. We then have that:

$$\frac{k}{c} + \frac{i^2}{ni}$$
$$= \frac{n^2 - 5ni + 4i^2}{n^2 - 4ni} + \frac{i}{n}$$

$$= \frac{n^2 - 5ni + 4i^2}{n(n-4i)} + \frac{i}{n}$$
$$= \frac{n^2 - 5ni + 4i^2 + (n-4i)(i)}{n(n-4i)}$$
$$= \frac{n^2 - 4ni}{n(n-4i)}$$
$$= 1.$$

By Lemma 3.1, we can decompose the edges between U and W' into $a = ni$ copies of S_k with each vertex of U a center of one copy of S_k and $c = n^2 - 4ni$ copies of $S_{z=i^2}$ with each vertex in W' a center of one copy of S_z.

Let D_{uw} be the set of S_{5ni-n^2}'s and $D_{uw'}$ be the set of S_k and D_{wu} be the set of S_{i^2}'s in this decomposition.

Again by Lemma 2.1, we can decompose the edges between V and W'' into ni copies of S_{5ni-n^2} with each vertex of V the center of one copy of S_{5ni-n^2}. We let $l = y_1 + y_2 - 5ni + n^2 = 8i^2 - 6ni + n^2$. Observe that l is positive for all $n \geq 4i$. We can then show that,

$$\frac{l}{c} + \frac{2i^2}{ni}$$
$$= \frac{n^2 - 6ni + 8i^2}{n^2 - 4ni} + \frac{2i}{n}$$
$$= \frac{n^2 - 6ni + 8i^2}{n(n-4i)} + \frac{2i}{n}$$
$$= \frac{n^2 - 6ni + 8i^2 + (n-4i)(2i)}{n(n-4i)}$$
$$= \frac{n^2 - 4ni}{n(n-4i)}$$
$$= 1.$$

By Lemma 3.1, there is a decomposition of the edges between V and W' into $b = ni$ copies of S_l and $c = n^2 - 4ni$ copies of S_{2i^2}.

Let D_{vw} be the set of S_{5ni-n^2}'s and $D_{vw'}$ be the set of S_l and D_{wv} be the set of S_{2i^2}'s in this decomposition.

We now let $D_u = D_{uv} \cup D_{uw} \cup D_{uw'}$; observe that each vertex in U is the center of one copy of S_{2i^2}, one copy of S_{5ni-n^2} and one copy of $S_{k=n^2-5ni+4i^2}$, the union of which gives us S_{6i^2}. By Lemma 2.1, we can then decompose each S_{6i^2} into two copies of S_{3i^2}.

Similarly, we let $D_v = D_{vu} \cup D_{vw'} \cup D_{vw''}$, observe that each vertex in V is the center of one copy of S_{ni-2i^2}, one copy of S_{5ni-n^2} and one copy of $S_{l=8i^2-6ni+n^2}$, the union of which gives us S_{6i^2}. Again, by Lemma 2.1, we can then decompose each S_{6i^2} into two copies of S_{3i^2}.

Finally, we let $D_w = D_{wu} \cup D_{wv}$, and note that each vertex in W' is the center of one copy of S_{i^2}, and one copy of S_{2i^2}, the union of which gives us S_{3i^2}.

Case 3.3.2.4 *$n \geq 5i$.*

Let $m = \frac{n-n'}{3i}$ where $2i < n' \leq 5i$. Observe that we can partition U into subsets U' and U'' such that $|U'| = n'i$ and $|U''| = 3mi^2$. Similarly, we can also partition V into V' and V'' and W into W' and W'' such that $|U'| = |V'| = |W'| = n'i$ and $|V''| = |U''| = |W''| = 3mi^2$. By Lemma 2.1, there is a S_{3i^2} decomposition of the edges between U'' and V', U'' and W', U'' and V'', U'' and W'', V'' and W', V'' and U', V'' and W'', W'' and U' , W'' and V'. The remaining edges that are not decomposed are the edges between each of U', V' and W', i.e. a graph isomorphic to $K_{n'i,n'i,n'i}$. We can then use cases 3.3.2.1, 3.3.2.2, 3.3.2.3 to decompose the remaining edges.

Case 3.3.3 *$r = i^2 j, p = nij$ with $p \geq r$.*

Observe that we can partition the edges of $K_{p,p,p}$ into the union of 3 subgraphs of $K_{p,p}$ and we can then use Theorem 2.3 to decompose the graph into S_r.

Case 3.3.4 *$r = i^2 j, p = nij$ with $\frac{2}{3}r \leq p \leq r$.*

Using Lemma 2.2, we can show that $K_{p,p,p}$ has an S_r decomposition if $K_{ni,ni,ni}$ has a S_i^2 decomposition. Observe that when $\frac{2}{3}r \leq p \leq r$, $\frac{2i}{3} \leq n \leq i$. Let U,V and W be the three partite sets of $K_{ni,ni,ni}$. Referring to case 3.3.2.1, we define an S_{i^2}-decomposition with the following values:

$$a = ni;$$

$$b = ni;$$

$$c = 3n^2 - 2ni;$$

$$x = 3n^2 - 3ni + i^2;$$

$$y = 3n^2 - 4ni + 2i^2;$$

$$z = ni.$$

We then assume there exists a S_{i^2}-decomposition where there are $a = ni$ copies of S_i^2 with each vertex of U a center of one copy of S_i^2 with x edges between partite set U and V and $i^2 - x$ edges between partite set U and W; $b = ni$ copies of S_i^2 with each vertex of V a center of one copy of S_i^2 with y edges between partite set V and W and $i^2 - y$ edges between partite set V and U; and c copies of S_i^2 with $c = 3n^2 - 2ni$ vertices of W a center of one copy of S_i^2 with z edges between partite set W and U and $i^2 - z$ edges between partite set W and V. We will now show that our choice above fulfils the requirements for such a decomposition to exist.

Referring to equations (3.1), (3.2), (3.3), we have

$$a(x) + b(i^2 - y)$$

$$= ni(3n^2 - 3ni + i^2) + ni(i^2 - (3n^2 - 4ni + 2i^2))$$

$$= ni(3n^2 - 3ni + i^2) + ni(4ni - i^2 - 3n^2)$$

$$= ni(3n^2 - 3ni + i^2 + 4ni - i^2 - 3n^2)$$

$$= n^2i^2;$$

$$b(y) + c(i^2 - z)$$

$$= ni(3n^2 - 4ni + 2i^2) + (3n^2 - 2ni)(i^2 - ni)$$

$$= ni(3n^2 - 4ni + 2i^2) + ni(3n - 2i)(i - n)$$

$$= ni(3n^2 - 4ni + 2i^2 + 5ni - 3n^2 - 2i^2)$$

$$= n^2i^2;$$

$$c(z) + a(i^2 - x)$$

$$= (3n^2 - 2ni)(ni) + ni(i^2 - (3n^2 - 3ni + i^2))$$

$$= ni(3n^2 - 2ni) + ni(3n^2 + 3ni$$

$$= ni(3n^2 - 2ni + 3n^2 + 3ni)$$

$$= n^2i^2.$$

From the description of the decomposition, the values of x,y and z are bound by the following inequalities:

$$i^2 - ni \leq x \leq ni;$$

$$i^2 - ni \leq y \leq ni;$$

$$i^2 - ni \leq z \leq ni.$$

From equation (3.1), we have

$$a(x) - b(i^2 - y) = n^2i^2$$

$$\Rightarrow ni(x) - ni(i^2 - y) = n^2i^2.$$

Dividing both sides by n^2i^2 gives us

$$\Rightarrow \frac{x}{ni} - \frac{3i^2 - y}{ni} = 1.$$

By Lemma 3.1, the edges between U and V can be decomposed into ni copies of S_x and ni copies of S_{i^2-y} so that each vertex of U is the center of one copy of S_x and each vertex of V is the center of one copy of S_{i^2-y}. Let D_{uv} be the set of S_x's and D_{vu} be the set of S_{i^2-y}'s in this decomposition.

We can partition the W into disjoint sets W' and W'', such that $|W'| = c$ and $|W''| = ni - c$.

Observe that $i^2 - x = ni - c$:

$$i^2 - (3n^2 - 3ni + i^2)$$

$$= 3ni - 3n^2$$

$$= ni + 2ni - 3n^2$$

$$= ni - c.$$

By Lemma 2.1 we can decompose the edges between U and W'' into ni copies of S_{i^2-x} with each vertex of U the center of one copy of S_{i^2-x}. By Lemma 2.1, we can also decompose the edges between U and W'' into c copies of S_{ni} with each vertex of W'' the center of one copy of S_{ni}. We let D_{uw} be the set of S_{i^2-x}'s and D_{wu} be the set of S_{ni}'s in this decomposition.

Again, by Lemma 2.1 we can decompose the edges between V and W'' into ni copies of S_{ni-c} with each vertex of V the center of one copy of S_{ni-c}. Observe that $y \geq ni - c$, the proof of which is as follows:

$$y \geq ni + c$$

$$\iff 3n^2 - 4ni + 2i^2 \geq ni - 3n^2 + 2ni$$

$$\iff 6n^2 - 7ni + 2i^2 \geq 0$$

$$\iff (2n - i)(3n - 2i) \geq 0$$

which is true since n $\geq \frac{2i}{3}$.

Also observe that $\frac{y-ni+c}{c} + \frac{i^2-z}{ni} = 1$, the proof of which is as follows:

$$\frac{3n^2 - 4ni + 2i^2 - ni + c}{c} + \frac{i^2 - z}{ni}$$

$$= \frac{3n^2 - 4ni + 2i^2 - ni}{3n^2 - 2ni} + 1 + \frac{i^2 - ni}{n}$$

$$= \frac{3n^2 - 5ni + 2i^2}{n(3n - 2i)} + \frac{i - n}{n} + 1$$

$$= \frac{3n^2 - 5ni + 2i^2 + (3n - 2i)(i - n)}{n(3n - 2i)} + 1$$

$$= \frac{3n^2 - 5ni + 2i^2 + 5ni - 2i^2 - 3n^2}{ni(n^2 - 2ni)} + 1 = 1$$

By Lemma 3.1 we have a decomposition between the edges of V and W' with ni copies of S_{y-ni+c} with each vertex of V the center of one copy of S_{y-ni+c} and c copies of S_{i^2-ni} with each vertex of W' the center of one copy of S_{i^2-ni}. Let $D_{vw''}$ be the set of S_{ni-c}'s, $D_{vw'}$ be the set of S_{y-ni+c}'s and D_{wv} be the set of S_{i^2-z}'s.

We now let $D_u = D_{uv} \cup D_{uw}$, observe that each vertex in U is the center of one copy of S_x and one copy of S_{i^2-x}, the union of which is S_{i^2}. Similarly, we let $D_v = D_{vu} \cup D_{vw'} \cup D_{vw''}$, each vertex in V is the center of one copy of S_{i^2-y}, one copy of S_{ni-c} and one copy of S_{y-ni+c}, the union of which is S_{i^2}. Finally, we let $D_w = D_{wu} \cup D_{wv}$, and note that each vertex in W' is the center of one copy of S_z, and one copy of S_{i^2-z}, the union of which gives us S_{i^2}. Figure 3.3 is an illustration of an S_{16}-decomposition of $K_{10,10,10}$.

□

3.3 S_3-Decomposition of complete tripartite graphs

Theorem 3.4 *The complete tripartite graph $K_{p,q,r}$ decomposes into S_3 if and only if one of the following conditions is true:*

i. at least two of $p, q,$ and r is divisible by 3.

ii. $pq + pr + qr$ is divisible by 3 and $p, q, r \geq 2$.

Proof. Observe that edges of $K_{p,q,r}$ is the union of the bipartite graphs $K_{p,q}$, $K_{p,r}$, and $K_{q,r}$. By the definition of a decomposition the number of edges in the decomposition has to divide the total number of edges in the graph, therefore $pq + pr + qr (\bmod 3) = 0$.

Let $p' = p \pmod 3$, $q' = q \pmod 3$, $r' = r \pmod 3$. We then construct a table for the values of $pq + pr + qr (\bmod 3)$.

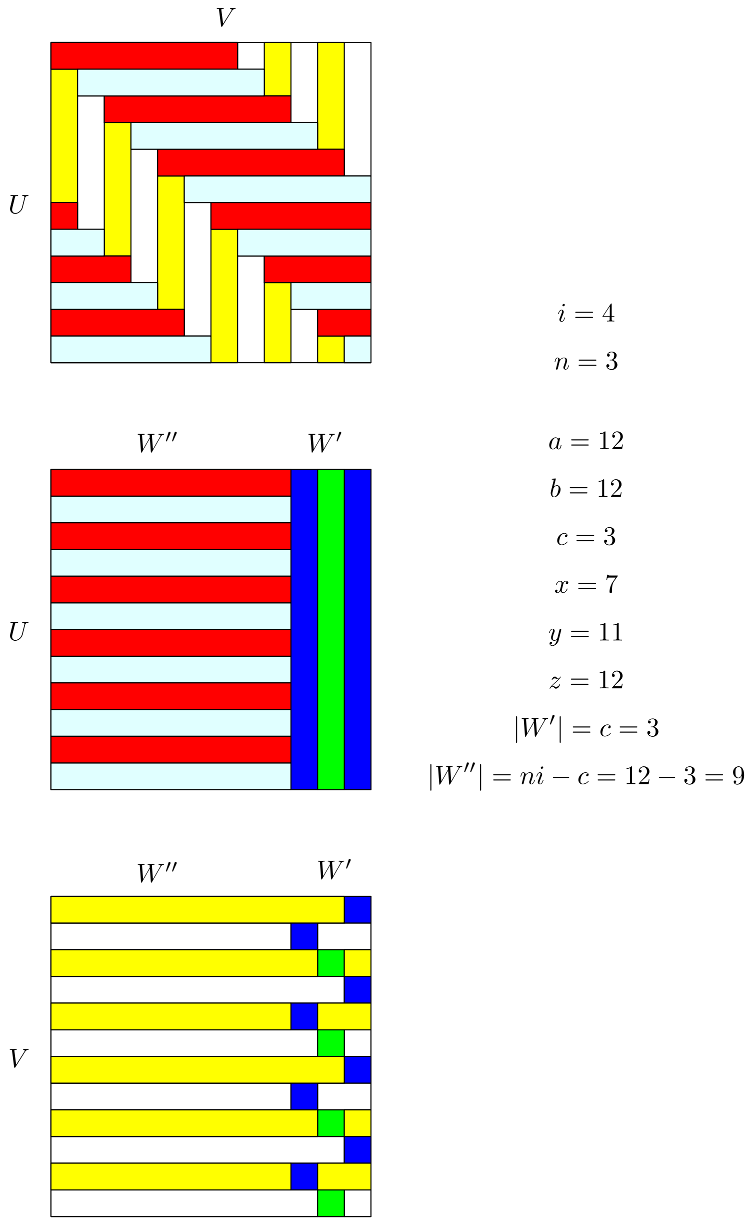


Figure 3.3: $K_{12,12,12}$ decomposed into S_{16}.

From Table 3.1, we can divide our proof into two separate cases. Observe that the statement of the first condition of Theorem 3.4 is equivalent to Case 3.4.1.

Case 3.4.1: At least two of p', q' and r' are equal to 0.

Case 3.4.2: $p' = q' = r' = d \neq 0$.

Case 3.4.1 *At least two of p', q' and r' are equal to* 0.

Without loss of generality let $p' = q' = 0$. Observe that $K_{p,q,r}$ is the union of the bipartite graphs $K_{p,q}$, $K_{p,r}$, and $K_{q,r}$. Observe that in each of

		q'		
	$r' = 0$	0	1	2
p'	0	0	0	0
	1	0	1	2
	2	0	2	1

		q'		
	$r = 1$	0	1	2
p'	0	0	1	2
	1	1	0	2
	2	2	2	2

		q'		
	$r' = 2$	0	1	2
p'	0	0	2	1
	1	2	2	2
	2	1	2	0

Table 3.1: The value $pq + qr + pr \pmod 3$ for different values of p', q' and r'.

the three bipartite graphs, there is at least one of the partite set with size divisible by three. By Lemma 2.1, we have an S_3 decomposition. Note that an S_3-decomposition exists when $r = 1$.

Case 3.4.2 $p' = q' = r' = d \neq 0$

Without loss of generality let $p \geq q \geq r$. Let U be the partite set with size p, V be the partite set with size q and W be the partite set with size r. Since $p \geq q \geq r$, $p = r + 3i$; $q = r + 3j$ for some $i, j \geq 0$. We then partition U into U' and U'' where $|U'| = r$ and $|U''| = 3i$, and we partition V into V' and V'' where $|V'| = r$ and $|V''| = 3j$.

Observe that by Lemma 2.1, we can partition the edges between U'' and V', U'' and V'', and U'' and W, and V'' and W into S_3 as the vertices in partite sets U'' and V'' of each subgraph has degree divisible by 3. The remaining edges that are not decomposed are the edges between partite sets U', V' and W. Observe that these edges, are the edges of graph $K_{r,r,r}$ and from case 3.3.1 of Theorem 3.3, there is a S_3 decomposition if p, q and r is greater or equal to 2. □

3.4 Extending Theorem 3.3 for multipartite graphs

The results of Ushio, Tazawa, and Yamamoto [37] shows that there is a S_r-decomposition of a complete m-partite graph $K_{p,p,\ldots,p}$ if and only if $\binom{m}{2}p^2 \equiv 0 \pmod{r}$ and $mp \geq 2r$. In this section we discuss whether the methods of Theorem 3.3 can be generalized to proof the same result.

We found that as m becomes larger, the number of variables and subdivision of cases increases. The following is not an exhaustive construct to cover all possible decompositions. We outline a proof in the case $3i \leq n \leq 5i$, $m = 4$. Let $r = 6i^2j$; observe that we can obtain a S_{i^2j}, S_{2i^2j} and S_{3i^2j}-decomposition from a S_{6i^2j}-decomposition. We then have the following:

$$r \mid 6p^2$$

$$\Rightarrow i^2j \mid p^2$$

$$\Rightarrow ij \mid p$$

$$\Rightarrow p = nij$$

By Lemma 2, there exists a S_r-decomposition of $K_{p,p,p,p}$ if there is a S_{6i^2}-decomposition of $K_{ni,ni,ni,ni}$.

We let T, U, V, W be the 4 partite sets of $K_{ni,ni,ni,ni}$. We define the decomposition using by using the definition set in table 3.2.

	T	U	V	W
number of centers in the partite set	a	b	c	d
Number of edges to Partite set T		t_1	t_2	t_3
Number of edges to Partite set U	u_1		u_2	u_3
Number of edges to Partite set V	v_1	v_2		v_3
Number of edges to Partite set W	w_1	w_2	w_3	

Table 3.2: Table describing the S_{6i^2} decomposition.

Observe that from Table 3.2, $t_1 + t_2 + t_3 = u_1 + u_2 + u_3 = v_1 + v_2 + v_3 = w_1 + w_2 + w_3 = 6i^2 \rightarrow (1.1)$ is a necessary condition for this construction to be a S_{6i^2}-decomposition.

We then assume there exists a S_{6i^2}-decomposition such that there are $a = ni$ copies of S_{6i^2} with each vertex of T a center of one copy of S_{6i^2} with $t_2 = ni$ edges between partite set T and V and $t_3 = 6i^2 - ni$ edges between partite set T and W. We also assume that there are $b = ni$ copies of S_{6i^2} with each vertex of U a center of one copy of S_{6i^2} with $u_1 = ni$ edges between partite set T and U and $u_3 = 6i^2 - ni$ edges between partite set U and W. We assume that there are $c = ni$ copies of S_{6i^2} with each vertex of V a center of one copy of S_{6i^2} with $v_2 = ni$ edges between partite set U and V and $v_3 = 6i^2 - ni$ edges between partite set V and W. Finally, we assume that there are $d = n^2 - 3ni$ vertices selected as centers of S_{6i^2} in partite set W, with $2i^2$ edges to partite set T, $2i^2$ edges to partite set U, $2i^2$ edges to partite set V. This decomposition is summarized in Table 3.3.

	T	U	V	W
number of centers in the partite set	ni	ni	ni	$n^2 - 3ni$
Number of edges to Partite set T		0	ni	$6i^2 - ni$
Number of edges to Partite set U	ni		0	$6i^2 - ni$
Number of edges to Partite set V	0	ni		$6i^2 - ni$
Number of edges to Partite set W	$2i^2$	$2i^2$	$2i^2$	

Table 3.3: Table of values for S_{6i^2}-decomposition for graph $K_{ni,ni,ni,ni}$.

By considering the edges between each pair of the partite sets we have,

$$a(t_1) + b(u_1) = n^2i^2;$$

$$a(t_2) + c(v_1) = n^2i^2;$$

$$a(t_3) + d(w_1) = n^2i^2;$$

$$b(u_2) + c(v_2) = n^2i^2;$$

$$b(u_3) + d(w_2) = n^2i^2;$$

$$c(v_3) + d(w_3) = n^2i^2. \tag{3.14}$$

The construction of the decomposition also gives us the following bounds,

$$0 \leq t_1, t_2, t_3 \leq ni;$$

$$0 \leq u_1, u_2, u_3 \leq ni;$$

$$0 \leq v_1, v_2, v_3 \leq ni;$$

$$0 \leq w_1, w_2, w_3 \leq ni. \tag{3.15}$$

We let,

$$a = b = c = ni;$$

$$w_1, w_2, w_3 = 2i^2;$$

$$t_2 = u_1 = v_2 = ni;$$

$$t_1 = u_2 = v_1 = 0;$$

$$t_3 = u_3 = v_3 = 6i^2 - ni;$$

$$d = n^2 - 3ni. \tag{3.16}$$

We will now show that our choice fulfils equations (3.14),

$$a(t_1) + b(u_1)$$

$$= ni(0) + ni(ni) = n^2i^2;$$

$$a(t_2) + c(v_1) = n^2i^2$$

$$= ni(ni) + ni(0) = n^2i^2;$$

$$a(t_3) + d(w_1) = n^2i^2$$

$$= ni(6i^2 - ni) + (n^2 - 3ni)(2i^2)$$

$$= ni(6i^2 - ni) + (ni)(n - 3i)(2i)$$

$$= ni(6i^2 - ni - 6i^2 + 2ni)$$

$$= n^2i^2.$$

Since our choice is symmetric, it is not difficult to see that the rest of the equations are also satisfied. We also note that our choice of the values fulfils the bounds given in inequalities 3.15.

We verify that the sum of the edges totals $6i^2$ as required in condition (1.1).

$$t_3 + t_2 + t_1 = u_1 + u_2 + u_3 = v_1 + v_2 + v_3$$

$$= ni + 6i^2 - ni + 0 = 6i^2.$$

By Lemma 2.1, we can decompose the edges between T and V, using ni copies of S_{ni} so that each of the vertex in partite set T is a center of a copy of S_{ni}. We D_t be the set of S_{ni} in this decomposition.

Similarly, we can decompose the edges between T and U using ni copies of S_{ni} so that each vertex in partite set U is a center of a copy of S_{ni}. We let D_u be the set of S_{ni} in this decomposition. Finally, we can decompose the edges between U and V using ni copies of S_{ni} so that each of the vertex in partite set V is a center of a copy of S_{ni}. We let D_v be the set of S_{ni} in this decomposition.

Let $d' = n^2 - (3+k)ni$ where $k =$ floor$(n/i - 3)$. We partition W into two disjoint subsets W' and W'' where $|W'| = ni - d'$ and $|W''| = d'$. Let $x = (n - 2(k+1)i)(n - (3+k)i) = 6i^2 + 8ki^2 - 3kin + 2k^2i^2 - 5ni + n^2$.

Observe that $\frac{6i^2-ni-x}{ni-d'} + \frac{2ki^2}{ni} = 1$, the proof of which is as follows:

$$= \frac{6i^2 - ni - (6i^2 + 8ki^2 - 3kin + 2k^2i^2 - 5ni + n^2)}{((4+k)ni - n^2)} + \frac{2ki^2}{ni}$$

$$= \frac{4ni - 8ki^2 - 2k^2i^2 + 3kin - n^2)}{((4+k)ni - n^2)} + \frac{2ki}{n}$$

$$= \frac{2kin - 8ki^2 - 2k^2i^2}{n((4+k)i - n)} + 1 + \frac{2ki}{n}$$

$$= \frac{2kin - 8ki^2 - 2k^2i^2 + ((4+k)i - n)(2ki)}{n((4+k)i - n)} + 1$$

$$= \frac{2kin - 8ki^2 - 2k^2i^2 + (8ki^2 + 2k^2i^2 - 2kin)}{n((4+k)i - n)} + 1$$

$$= 1.$$

Also observe that $\frac{x}{d'} + \frac{(k+1)2i^2}{ni} = 1$, the proof of which is as follows:

$$\frac{x}{d'} + \frac{(k+1)2i^2}{ni}$$

$$= \frac{(n - 2(k+1)i)(n - (3+k)i)}{n^2 - (3+k)ni} + \frac{2(k+1)i^2}{ni}$$

$$= \frac{(n - 2(k+1)i)}{n} + \frac{2(k+1)i}{n}$$

$$= 1.$$

By Lemma 3.1, we can decompose the edges between T and W' using $a = ni$ copies of S_{6i^2-ni-x} with each vertex of T as the center of a copy of

S_{6i^2-ni-x} and $ni-d'$ copies of S_{2ki^2} with each vertex of W' a center of a copy of S_{2ki^2}. We can also decompose the edges between T and W'' using a copies of S_x with each vertex of T as the center of a copy of S_x and d' copies of $S_{(2k+2)i^2}$ with each vertex of W'' a center of a copy of $S_{(2k+2)i^2}$. Let $D_{tw'}$ be the set of S_{6i^2-ni-x} and $D_{tw''}$ be the set of S_x in this decomposition. Let $D_{wt'}$ be the set of S_{2ki^2} and $D_{wt''}$ be the set of $S_{(2k+2)i^2}$ in this decomposition.

Let $D_T = D_t \cup D_{tw'} \cup D_{tw''}$. Observe that each vertex in U is the center of one copy of S_{ni}, one copy of S_{6i^2-ni-x} and one copy of S_x, the union of which gives us S_{6i^2}.

Observe that since $a=b=c$ and $t_3=u_3=v_3$ and $w_1=w_2=w_3$ the edges between U and W and the edges between V and W decompose in the same manner as the decomposition described for T and W. Since we have that the decomposition between W and the other two partite sets are identical, each vertex in W' is the center of three copies of S_{2ki^2}. We can then rearrange the decomposition such that each vertex in W' is the center of k copies of S_{6i^2}. Similarly, observe that each vertex in W'' is the center of three copies of $S_{(2k+2)i^2}$. We can also rearrange the decomposition such that each vertex in W'' is the center of $k+1$ copies of S_{6i^2}.

What we have done here works for the case $3i \leq n \leq 5i$. Note that x is necessarily positive, therefore for $n \geq 5i$ we have an obstacle. For these cases, we may need to introduce a second star on one of the partite sets as in the Case 3.3.2.3 to obtain a S_{6i^2}-decomposition. For the cases where $7i \leq n \leq 9i$ we may use the strategy in Case 3.3.2.2 to reduce the case to $3i \leq n' \leq 5i$. Moreover, for the cases where $9i \leq n \leq 11i$ we may use the strategy in Case 3.3.2.4 to reduce the case to $3i \leq n' \leq 5i$.

We now discuss the case where there are more than four partite sets, i.e. $m > 4$. As a general rule, the algorithm detailed here and in Theorem 3.3, the S_r-decomposition of $K_{ni,ni,\ldots,ni}$ works best if we choose $m-1$ partite sets to be the centers of kni copies of S_r. Observe that when Lemma 3.2 is extended to m-partite graphs, it is necessary that every vertex of $m-1$ partite sets are centers of at least one copy of S_r. Moreover, choosing every vertex of $m-1$ partite sets to be centers of k copies of S_r reduces the number of partitions needed on the partite sets and hence makes it simpler to ensure that the necessary conditions for Lemma 3.1 are met. The remaining number of centers of S_r for the partite set (we call this partite set X) that is not an ni-multiple, would then by construction, have the number of vertex used as the centers of a S_r being a multiple of n.

From here, we may choose a multiple of i for the number of edges between partite set X and the other partite sets. This helps ensures that we can obtain integer solutions for equations (3.14). Finally, it is important to check that the values selected are within the bounds given in (3.15). It may be necessary to make each partite set the center of multiple copies of S_r as in Case 3.3.2.3 if the bounds are not satisfied. Observe also that for larger values of n, we may be able to reduce the case using methods detailed in Case 3.3.2.2 and Case 3.3.2.4.

Chapter 4

Decomposition of regular bipartite Graphs

In this chapter we study the decomposition of d-regular bipartite graphs into S_r. In particular, we will discuss various strategies for the decomposition of 4-regular bipartite graphs into S_3 as a base case for the decomposition of other d-regular bipartite graphs. In order to impose additional structure to the bipartite graphs, we will study different strategies firstly on a class of bipartite graphs discussed in the introduction section of this thesis as cyclic bipartite graphs. For notation, we let $B_{n,n}$ be a 4-regular cyclic bipartite graph with n vertices on two partite sets labelled as U and V. While we have introduced $O(1)$ algorithms for the decompositions in the earlier sections, this decomposition problem has been conjectured to be NP-complete [24].

4.1 S_3-decomposition of 4-regular bipartite graphs

4.1.1 Strategy 1: Picking one edge from each vertex in one partite set to form S_3.

Let U and V be the two partite sets of G where G is a 4-regular bipartite graph. Observe that the two partite sets of the bipartite graph are identically sized. Let $|U| = |V| = n$. Observe that the number of edges in G is $4n$. By the definition of a decomposition $4n$ must divide 3, and therefore n is necessarily divisible by 3.

For our initial analysis, we will look into a special class of 4-regular bipartite graph that is said to be 'cyclic', as defined in the introduction. We let $U = \{u_0, u_1, u_2 \ldots u_n\}$ and $V = \{v_0, v_1, v_2 \ldots v_n\}$. We let $D = \{d_0, d_1, d_2, d_3\}$, where $d_0 < d_1 < d_2 < d_3 < n$ as the generator set D such that u_i is adjacent to v_j if and only if $i + d_k (\bmod n) = j$ for some $d_k \in D$.

Next, observe that, if we delete one edge from every vertex in V, then every vertex in the partite set V has degree 3, and by Lemma 2.1, we can decompose the remaining edges into copies of S_3. Hence, if we can form $\frac{n}{3}$ copies of S_3 using $\frac{n}{3}$ vertices of partite set U as the center of one copy of S_3, such that each vertex of V is used exactly once, we can say that there is a S_3-decomposition of the graph G. We say that such a set of graphs is an S_3-cover for V. Figure 4.1 gives an illustration of this strategy. Observe that every vertex in the partite set on the right has degree 3.

In our analysis, we found that we can reduce the number of test cases, without losing generality. First, we can assume that the first difference d_0 is 0, otherwise we can subtract every element of the generator set D by d_0. Second, we can assume that the difference between $d_3 - d_0 \pmod n = d_3$ is not greater than $d_0 - d_1 \pmod n$, $d_1 - d_2 \pmod n$ and $d_2 - d_3 \pmod n$, otherwise we can reorder the generator set. Observe that $d_3 - d_0 \leq \frac{3n}{4}$. The proof of which is as follows:

We assume for the sake of contradiction that $d_3 - d_0 (\bmod n) > \frac{3n}{4}$. Since $d_0 = 0$ and $d_3 < n$, this assumption also gives us $d_3 > \frac{3n}{4}$. Since we have $d_3 > d_2 > d_1 > d_0$, we can derive the following inequalities:

$$d_2 - d_3 \pmod n \geq d_3 > \frac{3n}{4},$$
$$\Rightarrow d_2 - d_3 + n > \frac{3n}{4},$$
$$\Rightarrow d_2 > d_3 - \frac{n}{4};$$

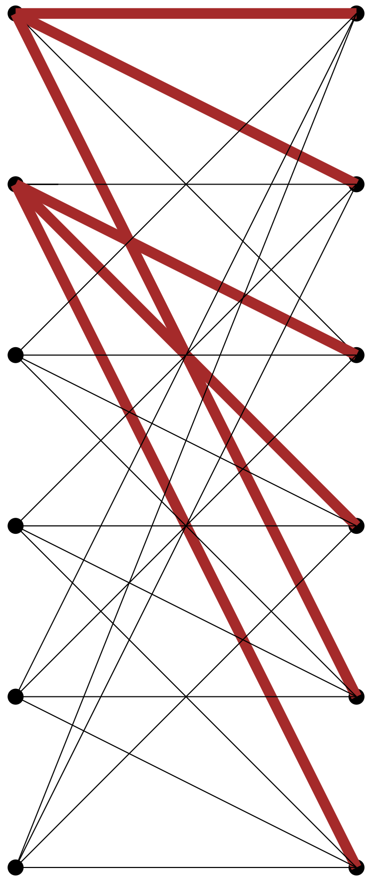


Figure 4.1: S_3 decomposition of a 4-regular graph using Strategy 1

$$d_1 - d_2 \pmod n \geq d_3 > \frac{3n}{4},$$
$$\Rightarrow d_1 - d_2 + n > \frac{3n}{4},$$
$$\Rightarrow d_1 > d_2 - \frac{n}{4};$$

$$d_0 - d_1 \pmod n \geq d_3 > \frac{3n}{4},$$
$$\Rightarrow d_0 - d_1 + n > \frac{3n}{4},$$
$$\Rightarrow 0 - d_1 > -\frac{n}{4},$$
$$\Rightarrow d_1 < \frac{n}{4}.$$

By combining the inequalities, we then find a contradiction on d_3,

$$\frac{n}{4} > d_1 > d_2 - \frac{n}{4},$$

$$d_2 < \frac{n}{2};$$

$$\frac{n}{2} > d_2 > d_3 - \frac{n}{4},$$

$$d_3 < \frac{3n}{4}.$$

Observe also that the difference between two successive elements of D is less or equal to $n - d_3$. Finally, we can assume that vertex u_0 is always picked as the center of a copy of S_3.

Let $\mu(xy)$ is the number of edges between x and y. In the case of a simple graph, $\mu(xy) = 1$ if and only if x is adjacent to y. Let $c(x)$ be the center function on x, where $c(x)$ is the number of copies of S_k with x as the center and let $|E(S)|$ be the number of edges in the subgraph induced by S.

Hoffman [23], stated that a star-design, exists for a graph G if and only if the following conditions are true,

$$\text{i. } k \sum_{v \in G} c(v) = |E(G)|,$$

ii. For all,

$$\{x, y\} \in \binom{G}{2}, \mu(xy) \leq c(x) + c(y)$$

iii. For all $S \subseteq V$,

$$k \sum_{v \in S} c(v) \leq |E(S)| + \sum_{x \in S, Y \in G/S} \min(c(x), \mu(xy)).$$

We apply the above result to the strategy outlined above. Note that each vertex in V is a center exactly once, and $\frac{n}{3}$ vertices of U are centres exactly once and the remaining vertices are not centers.

Condition 1 is trivially true by the definition of a decomposition. By the construction of our strategy, every vertex in V is a center of a star S_3, and since every edge of a bipartite graph is between partite sets U and V, condition 2 is trivially true as well. We then use the condition 3 to find copies of S_3 which use each vertex from V exactly once.

Observe that condition 3 is most restrictive when S is the subset containing only the centers of S_k. Observe that $|E(G)| = 4n$. Observe also that S contains n vertices in partite set V and $\frac{n}{3}$ vertices in partite set U, therefore we have that $|E(S)| = \frac{4n}{3}$.

We then have,

$$k \sum_{v \in S} c(v) \leq |E(S)| + \sum_{x \in S, y \in V \setminus S} \min(c(x), \mu(xy))$$

$$\Rightarrow |E(G)| \leq |E(S)| + \sum_{x \in S, y \in G \setminus S} \min(c(x), \mu(xy))$$

$$\Rightarrow 4n \leq \frac{4n}{3} + \sum_{x \in S, y \in G \setminus S} \min(c(x), \mu(xy))$$

$$\Rightarrow \frac{8n}{3} \leq \sum_{x \in S, y \in G \setminus S} \min(c(x), \mu(xy)).$$

Observe that the number of edges between S and $G \setminus S = 4(n - \frac{n}{3})$. Therefore $\sum_{x \in S, Y \in G \setminus S} \mu(xy) = \frac{8n}{3}$. It is necessary that $c(x) \neq 0$ (i.e x is a center) for every edge $\{x, y\}$ where $x \in S$ and $y \in G \setminus S$, otherwise the inequality above is violated. From here, we say that the graph is "feasible" if and only if, every edge that is between S and the $V \setminus S$ includes a center of S_3.

Trivially, this condition is necessary, but Hoffman's result tells us this is sufficient which aids us greatly in finding a decomposition by computer.

Using these generalization, and the algorithms detailed by Hoffman, we wrote a simple JAVA program to find S_3-covers of the vertices in partite set V, (source code is in Appendix 6.1). While the program is able to solve for size $n \leq 30$ within a reasonable amount of time (under 1 second per generator set, 10 minutes for the results for all the possible generator sets)), the runtime increases exponentially with the number of vertices in the partite sets. It takes approximately 1 day for the results for $n = 42$ and an estimated 1 week for the results for $n = 45$. Output of the program for $n \leq 18$ is given in Appendix 6.2.

Table 4.1 is a sample output of the programme for $n = 9$.

Generator Set	Star 1	Star 2	Star 3
$\{0,1,2,3\}$	$(u_0; v_0, v_1, v_2)$	$(u_2; v_3, v_4, v_5)$	$(u_5; v_6, v_7, v_8)$
$\{0,1,2,4\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_3, v_5)$	$(u_6; v_6, v_7, v_8)$
$\{0,1,3,4\}$	$(u_0; v_0, v_1, v_3)$	$(u_2; v_2, v_5, v_6)$	$(u_4; v_4, v_7, v_8)$
$\{0,2,3,4\}$	$(u_0; v_0, v_2, v_3)$	$(u_1; v_1, v_4, v_5)$	$(u_4; v_6, v_7, v_8)$
$\{0,1,2,5\}$	$(u_0; v_0, v_1, v_5)$	$(u_2; v_2, v_3, v_4)$	$(u_6; v_6, v_7, v_8)$
$\{0,1,3,5\}$	$(u_0; v_0, v_1, v_5)$	$(u_1; v_2, v_4, v_6)$	$(u_7; v_3, v_7, v_8)$
$\{0,1,4,5\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_5, v_6)$	$(u_3; v_3, v_7, v_8)$
$\{0,2,3,5\}$	$(u_0; v_0, v_2, v_3)$	$(u_1; v_1, v_4, v_6)$	$(u_5; v_5, v_7, v_8)$
$\{0,2,4,5\}$	$(u_0; v_0, v_2, v_4)$	$(u_1; v_1, v_3, v_6)$	$(u_3; v_5, v_7, v_8)$
$\{0,3,4,5\}$	$(u_0; v_0, v_3, v_4)$	$(u_2; v_2, v_6, v_7)$	$(u_5; v_1, v_5, v_8)$
$\{0,1,3,6\}$	$(u_0; v_0, v_1, v_6)$	$(u_1; v_2, v_4, v_7)$	$(u_2; v_3, v_5, v_8)$
$\{0,1,4,6\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_5, v_7)$	$(u_2; v_3, v_6, v_8)$
$\{0,2,3,6\}$	$(u_0; v_0, v_2, v_6)$	$(u_1; v_1, v_3, v_7)$	$(u_2; v_4, v_5, v_8)$
$\{0,2,4,6\}$	$(u_0; v_0, v_2, v_4)$	$(u_1; v_3, v_5, v_7)$	$(u_4; v_1, v_6, v_8)$
$\{0,2,5,6\}$	$(u_0; v_0, v_2, v_5)$	$(u_1; v_1, v_3, v_6)$	$(u_2; v_4, v_7, v_8)$
$\{0,3,4,6\}$	$(u_0; v_0, v_3, v_4)$	$(u_1; v_1, v_5, v_7)$	$(u_2; v_2, v_6, v_8)$
$\{0,3,5,6\}$	$(u_0; v_0, v_3, v_5)$	$(u_1; v_1, v_4, v_6)$	$(u_2; v_2, v_7, v_8)$

Table 4.1: S_3-cover of Partite Set V for $n = 9$

Extending Strategy 1

Hoffman proved the necessity of the conditions above in Section 4 [23], by building a network of the design and by evaluating the flow capacity of the network. By calculating the flow capacity of the min-cut-max-flow network, and orienting the edges such that the each edge of S_k flows to from the center to the leaves, Hoffman then states that there is an S_k design on graph G, or equivalently graph G has a S_k-decomposition if and only if $f(e_{xy}) = \mu(xy)$ where $f(e_{xy})$ is total number of edges with ends x and y that are orientated from x to y, or equivalently, all the edges of the graph belong to S_k. Unfortunately, most polynomial time algorithms for min-cut-max-flow such as Ford-Fulkerson algorithm, allows for $f(e_{xy}) \leq \mu(xy)$. This problem is NP-Complete according to Chekuri, Khanna and Shepherd [10].

We suggest that this problem may be solvable by computer using mathematical optimizer software such as CVX [21]. We propose that we can model the flow in as in Figure 4.2. From there, we can define an objective function, such that the function is minimum when either 0 or 3 edges is selected for each vertex of U. This is a modification of the Ford-Fulkerson algorithm used to the maximum matching in bipartite graphs [27].

Using our program, we found that most (more than 90%) 4-regular cyclic bipartite graphs have S_3-decompositions. We managed to find certain classes of graphs with no S_3-decomposition. One such case are graphs with two or more components. We can quickly determine a graph with this property by checking for a value of k such that k divides d_0, d_1, d_2, d_3 and n. If there exists a $k > 1$, graph would then have k components, with each component isomorphic to a 4-regular cyclic bipartite with $\frac{n}{k}$ vertices in each partite set, and generator set $D' = \{\frac{d_0}{k}, \frac{d_1}{k}, \frac{d_2}{k}, \frac{d_3}{k}\}$. We can then check if $\frac{n}{k}$ divides 3. If this is not true, we conclude that the number of edges in each component is not divisible by 3, hence the graph has no S_3-decomposition. Otherwise, we refer to the results of $n' = \frac{n}{k}$ and $D = D'$.

We found that Strategy 1 failed to give an S_3 decomposition for a single component cyclic bipartite graph for $n \leq 39$ in two specific test cases. These two cases are,

i. $n = 15, D = \{0, 1, 3, 7\}$ labelled here as $G1$,

ii. $n = 15, D = \{0, 4, 6, 7\}$ labelled here as $G2$.

We note that $G1$ and $G2$ are isomorphic to each other, with the partite sets U and V swapped. We developed Strategy 2 after analysing this case. Strategy 2 successfully generated S_3-decompositions of $G1$ and $G2$.

4.1.2 Strategy 2: Reducing the number of vertices to be covered.

The general idea behind Strategy 2 is to reduce the number of vertices in partite set V that need to be covered with S_3. Strategy 2 assumes that there is no common difference between successive elements of D (i.e, $d_1 - d_0 = d_2 - d_1 = d_3 - d_2$ is not true).

Without loss of generality, we assume that the four vertices in U adjacent to v_0 are each centers of one copy of S_3. We label these four center vertices as u_0, u_1, u_2, and u_3.

Next, we choose eight distinct vertices of $V \setminus v_0$ that are adjacent to u_0, u_1, u_2, and u_3. Observe that this is possible only if there are no common difference between the successive elements of D, otherwise there will only be six distinct vertices. We label these vertices as $\{v_i : 1 \leq i \leq 8\}$. We then delete all four edges incident to v_0, and we choose eight distinct edges between u_i and v_j where $0 \leq i \leq 3$ and $1 \leq j \leq 8$, such that each v_j has one edge deleted, and each u_i has two edges deleted.

Observe that v_0 has no edges, and v_j, $1 \leq j \leq 8$ has degree 3, and by Lemma 2.1, we can decompose the edges incident to v_j into S_3. We then use Strategy 1, to delete $n - 9$ edges between the unlabelled vertices of U and V such that each unlabelled vertex of U has either three edges deleted or no edges deleted, and each unlabelled vertex of V has one edge deleted. It may be necessary to choose a different set of eight vertices if we are unable to do the deletion with the unlabelled vertices of U and V.

Observe that the remaining edges are incident to the unlabelled vertices of V, and each of these vertex has degree 3. We then have a S_3 decomposition by Lemma 2.1.

We found that Strategy 2 is generally easier to do by hand for cases $n \leq 18$ but becomes extremely tedious when $n > 18$. It may be worthwhile to see the results of this strategy still holds when $n > 18$ using computers.

Figures 4.3, and 4.4 show the decomposition of $G1$ ($n = 15, D = \{0, 1, 3, 7\}$) and $G2$ ($n = 15, D = \{0, 4, 6, 7\}$) using Strategy 2.

4.1.3 Structure of a cyclic bipartite graph

Another strategy we tried was converting the graph into a line graph and observing the geometry. Let G be a connected 4-regular bipartite graph with partite sets U and V with size n where n is divisible by 3. Our initial observation yielded the following properties for $L(G)$:

a) there are $4n$ vertices in $L(G)$, and $12n$ edges in $L(G)$.

b) every vertex of $L(G)$ has degree 6. (This comes from the fact that G is 4 regular, and each vertex of $L(G)$ would then belong to 2 cliques of size 4).

c) We can partition the edges into 2 disjoint subsets $E1, E2$, such that every $v \in V(L(G))$ is common to exactly one pair of $\{e_i, e_j\}$ $e_i \in E1, e_j \in E2$. We can do this by choosing the elements of $E1$ to be the edges created from the vertices in U and the elements of $E2$ to be the edges created from the vertices in V.

We find that we can always factor $L(G)$ into P_2, because $L(G)$ is Hamiltonian and the number of vertices in $L(G)$ is divisible by 3. We can just group the vertices of $L(G)$ into groups of three along the Hamilton cycle. However a P_2-factor is insufficient to show that the G has a S_3-decomposition. We observed that if we can constraint the factors such that for every copy of $H = P_2, E(H) = \{e_i, e_j\}$, if we have $e_i, e_j \in E1$ or $e_i, e_j \in E2$, then we have an S_3-decomposition of G.

One advantage of using this method is that we have a visual representation of the decomposition problem. It is then more intuitive to find decompositions visually. Figure 4.5 illustrates how we may use the graph for this purpose. Note, we removed the edges between the cliques and replaced them with a line for clarity purposes.

The results of strategies 1, 2 and 3, obtained through our computer program showed that there is an S_3-decomposition for all cyclic 4-regular bipartite graphs with one component with size $n \leq 42$ if and only if n is divisible by 3. Cyclic 4-regular bipartite graphs with k components and size $n \leq 42$ have an S_3-decomposition if and only if n divides 3 and k is not divisible by 3. If k is divisible by 3, then the graph has an S_3-decomposition if and only if n is divisible by 9.

4.2 Probabilistic method on decomposition of bipartite graphs

In this section we discuss the results of Yuster [39] on tree decompositions and whether the results might be improved when applied to S_k-decompositions of bipartite graphs.

We say that a graph has property $P(H)$ if the necessary conditions for a $H-$decomposition is satisfied, namely, $|E(H)|$ divides $|E(G)|$ and $\gcd(H)$ divides $\gcd(G)$. Since H is a star, $\gcd(H) = 1$ and $\gcd(H)$ divides $\gcd(G)$ is trivially satisfied. Thus, $P(H)$ is reduced to $|E(H)|$ divides $|E(G)|$.

We let n be the number of vertices in G and h be the number of vertices in H. The star can then be denoted as S_{h-1}.

In the wording of Yuster, we define the problem statement as follows. Determine $f_H(n)$, the smallest possible integer, such that whenever G has n vertices and $\delta(G)$ (the minimum degree of G)$\geq fH(n)$, and G has property $P(H)$, then G also has a H-decomposition.

By Lemma 2.1 $f_H(n)$ is necessarily greater or equal to $h-1$. Using the example provided by Yuster as a guide, we can also show that for bipartite graphs, $f_H(n) > \frac{n}{4} - 1$. Consider a graph G where $n = 4x \geq 4h$, and $E(H)$ divides $2x^2$. Let G be 2 vertex-disjoint $K_{x,x}$ labelled here as G_1 and G_2. G has n vertices and $\delta(G) = x$. Since $x > h-1$, by Theorem 2.3 the condition $h-1 \mid x^2$ is the sufficient for a S_{h-1}-decomposition. If $h-1$ does not divide x^2 then we are done, otherwise we delete 1 edge from G_1 and $h-2$ edges from G_2. The resulting graph with minimum degree $x-1$, and $h-1$ divides $E(G)$ but G does not have a H-decomposition.

When G is a bipartite graph, we can tighten the bounds for an edge expanding graph in Theorem 1 [39]. Here, Yuster states that a graph with minimum degree $\delta(G) \geq \frac{n}{2} + r$ is also r-edge expanding. We can show that for a bipartite graph G, a graph with minimum degree $\delta(G) \geq \frac{n}{4} + r$ is r-edge expanding.

In the wording of Yuster, a graph is r-edge expanding if for every non-empty $X \subset V$ and $|X| \leq \frac{|V|}{2}$ there are at least $r|X|$ edges between X and $V \setminus X$. Consider a bipartite graph G. Let U_1 and U_2 be the partite sets of G. Let X_1 be m vertices of U_1 and X_2 be $|X| - m$ vertices of U_2. Let $X = U_1 \cup U_2$. Without loss of generality, let $m \leq \frac{|X|}{2}$, otherwise we swap partite sets. Observe that there are at most $m|X| - m^2$ edges between X_1 and X_2. Observe that there at least $(|X| - m)\delta(G)$ edges between X_2 and U_1. Observe also that there at least $(m)\delta(G)$ edges between X_1 and U_2.

Hence, there are at least

$$(|X| - m)\delta(G) + (m)\delta(G) - 2m(|X| - m)$$

$$= |X|\delta(G) - 2m(|X| - m)$$

edges between X and $V \setminus X$. We can show that $2m(|X| - m) \leq \frac{|X|^2}{2}$, the proof of which is as follows:

$$2m(|X| - m) \leq \frac{|X|^2}{2}$$

$$\iff 2m|x| - 2m^2 \leq \frac{|X|^2}{2}$$

$$\iff m|X| - m^2 \leq \frac{|X|^2}{4}$$

$$\iff m^2 - m|X| + \frac{|X|^2}{4} \geq 0$$
$$\iff (m - \frac{|X|}{2})^2 \geq 0$$

which is clearly true.

Since we have that $m \leq \frac{|X|}{2}$ and $|X| \leq \frac{|V|}{2}$, the number of edges between X and $V \setminus X$ is at least

$$|X|\delta(G) - 2m(|X| - m) \geq |X|\delta(G) - \frac{|X|^2}{2}$$
$$= |X|(\delta(G) - \frac{|X|}{2})$$
$$\geq |X|(\delta(G) - \frac{|V|}{4}).$$

Hence, a bipartite graph with $\delta(G) = \frac{|V|}{4} + r$ is also r-edge expanding.

Lemma 2.1 [39] states that if $G(V, E)$ is a graph with property $P(H)$, then E can be partitioned into $h - 1$ disjoint subsets $E_1, E_2, \ldots, E_{h-1}$ such that $|E_i| = m$ for $1 \leq i \leq h - 1$ and if the degree a vertex $v \in V$ in $G_i = (V, E_i)$ is denoted by $d_i(v)$, then for every $v \in V$, we have $|d_i(v) - \frac{d(v)}{h-1}| \leq 2.5\sqrt{d(v)logn}$, and each spanning subgraph G_i is $5h^3\sqrt{d(v)\log n}$.

Yuster constructed the proof by letting each edge $e \in E$ choose a random integer between 0 and $h-1$ where 0 is chosen with probability $\beta = n^{-\frac{1}{2}}$ and the other numbers are chosen with equal possibility $\alpha = \frac{1-\beta}{h-1}$. F_i for $0 \leq i \leq h - 1$ is defined as the set of edges which selected i. We observed that the expected value for the size of F_i, $E[|F_i|] = \alpha|E| = m(1 - \beta)$ for $i \neq 0$.

Yuster then defined $d'_i(v)$ as the number of edges adjacent to v which belongs to F_i. Note that the expected value for $d'_i(v) = \alpha d(v)$ for $1 \leq i \leq h - 1$ and $\beta d(v)$ for $i = 0$. Using the large Chernoff deviation [2], Yuster showed that with a probability greater than 0.9, we may obtain a “feasible” partition by transferring vertices from F_0 to F_i.

Lemma 2.2 states that a feasible orientation exists for every feasible partition of E. According to Yuster, an orientation is said to be Eulerian if the indegree and outdegree of every vertex differs by at most one. The existence of a feasible orientation is needed, as it defines a decomposition of the edges into

m sets L^* of edge-disjoint connected graphs where $m = \frac{|E(G)|}{h-1}$. Yuster defined $d_i^+(v)$ as the outdegree of v in E_i, and $d_i^-(v)$ as the indegree of v in E_i. Note that $d_i(v) = d_i^+(v) + d_i^-(v)$.

When H is a star, the orientation of the leaf vertices is trivially Eulerian, as the degree of every leaf vertex is 1. We can then obtain an Eulerian orientation by orienting the edges of adjacent to the center vertex such that $\lfloor \frac{h-1}{2} \rfloor$ edges are oriented away from the center vertex, and $\lceil \frac{h-1}{2} \rceil$ edges are oriented towards the center vertex.

Yuster's proof starts by selecting a leaf vertex using a breath first search algorithm (BFS), and labelling the vertex as q. He then select an edge from E_1, q is then selected to be a leaf of H, and is given an orientation such that q is the root of H. Observe that in the case of stars, the diameter of the tree is two. Hence, we have the following for Lemma 2.2 [39].

When $i = 1$, i.e. the edge adjacent to the leaf q. As in Yuster's result, we have the following,

$$|d_1^+(v) - d_1^-(v)| \leq 1 < 5\sqrt{n \log n}.$$

For $i = 2$, we have $j = p(2) = 1$.

$$|d_2^+(v) - d_2^-(v)| = |2c_v - d_i(v)| = |2d_1(v) - 2d_1^+ - d_2(v)|$$

$$\leq |2d_1^+ - d_1| + |d_1(v) - d_2(v)|$$

$$\leq |d_1^+ - d^1(v)| + |d_1(v) - \frac{d(v)}{h-1}| + |d_2(v) - \frac{d(v)}{h-1}|$$

$$\leq 1 + 5\sqrt{d(v) \log n}$$

$$\leq 5\sqrt{n \log n}.$$

Finally, when $3 \leq i \leq h-1$. Observe that v is a leaf of H, and $j = p(i) = 2$.

$$|d_i^+(v) - d_i^-(v)| = |2c_v - d_i(v)|$$

$$= |2d_2(v) - 2d_2^+ - d_i(v)|$$

$$\leq |2d_2^+ - d_2| + |d_2(v) - d_i(v)|$$

$$\leq |d_2^+ - d_2^-(v)| + |d_2(v) - \frac{d(v)}{h-1}| + |d_i(v) - \frac{d(v)}{h-1}|$$
$$\leq 5\sqrt{n \log n} + 5\sqrt{n \log n}$$
$$\leq 10\sqrt{n \log n}.$$

However, this improvement does not affect the overall result of Lemma 2.2 which states that in every feasible orientation, the outdegree $d_i^+ \geq 4h^3\sqrt{n \log n}$ for all $v \in V$ and for all $2 \leq i \leq h-1$. We give an outline of the proof for the rest of the paper.

Yuster states that every member of L^* is homomorphic to S_k, and every member that is a tree is isomorphic to H. Lemma 3.1 then states that, if all the perfect matching are selected randomly and independently, then with a probability of 0.9, there for all $0 \leq i \leq h-1$ and for all $v \in V(G)$, $|N(v,i)| \leq h\sqrt{(d_i^+(v))}$ where $N(v,i)$ are the neighbours of v in partition i.

Yuster then defined $L([u,j],[v,i])$ as the set of the members of L^* which contains an edge of $D_i^-(v)$ and an edge of D_j^-. Lemma 3.2 then showed that if the perfect matching are selected randomly and independently, then with a probability of 0.75, for every $u, v \in V(G)$ and for $0 \leq j < i \leq h-1$,—$L([u,j],[v,i])| \leq 2\sqrt{n \log n}$. Yuster then used the results of Lemma 3.1 and 3.2 to show that there is a probability of 0.65 that we can obtain a decomposition L^* with properties guaranteed by Lemma 3.1 and 3.2.

With the results of Lemma 3.1 and 3.2, Yuster then showed that we can mend L^* into a decomposition L consisting of only trees as the properties allows us to change the “bad” edges (defined here as edges that creates a cycle in L) with “good” edges.

Since the assumptions are unchanged, the results of Lemma 3.1 and Lemma 3.2 are therefore true, and we have that a $10h^4\sqrt{n \log n}$-edge expanding graph has a S_{h-1}-decomposition. We note that, it may be possible to tighten the bounds of the edge expansion by lowering the order of h. However as noted in equation (4) in Theorem 1, we require an $O(\sqrt{n \log n})$-edge expanding order, as a necessary condition for Lemma 3.2. Yuster conjectured that it may be possible to remove the requirements for an $O(\sqrt{n \log n})$-edge expansion factor, however we were unable show that we may remove the requirement is for S_{h-1}-decompositions of bipartite graphs.

With results above, we say that there is a S_{h-1}-decomposition for all bipartite graphs with a minimum degree $\delta(G) = \frac{n}{4} + 10h^4\sqrt{n \log n}$.

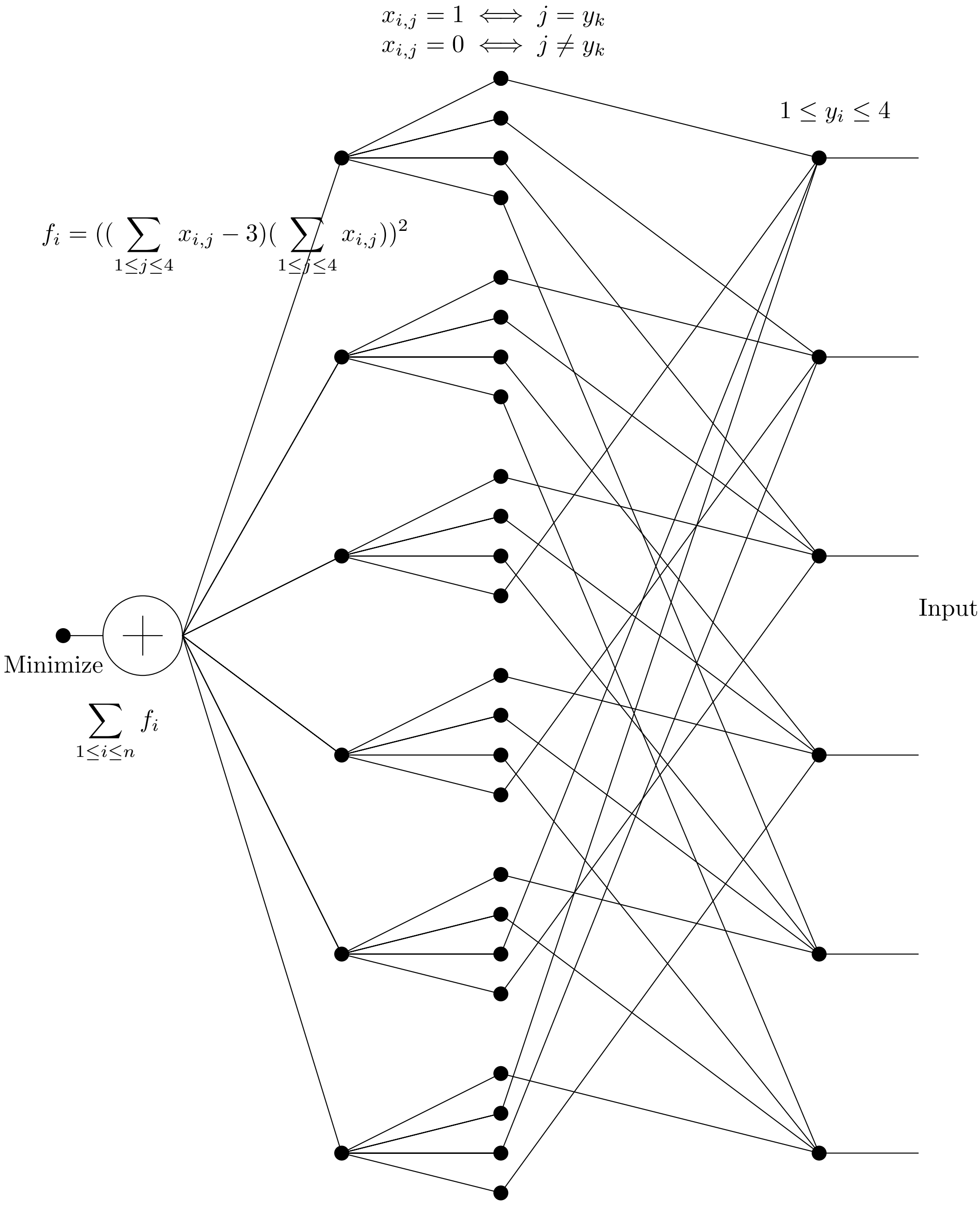


Figure 4.2: Using optimization software to find a S_3-cover of V.

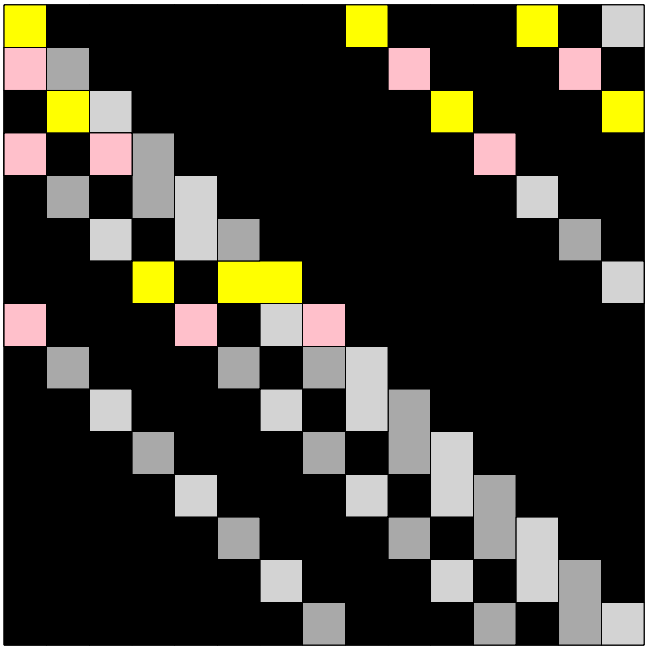


Figure 4.3: S_3-Decomposition of $G(n = 15, D = \{0, 1, 3, 7\})$; pink and yellow blocks are S_3 decompositions with centers in partite set U.

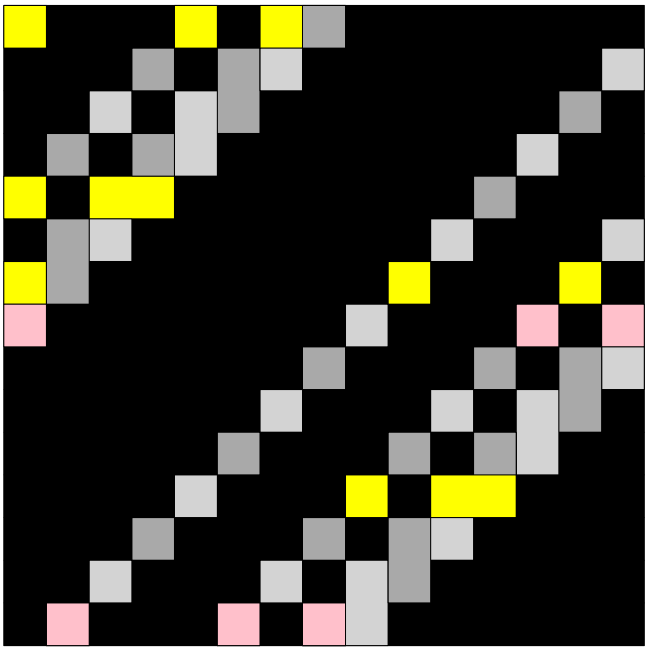


Figure 4.4: S_3-Decomposition of $G(n = 15, D = \{0, 4, 6, 7\})$; pink and yellow blocks are S_3 decompositions with centers in partite set U.

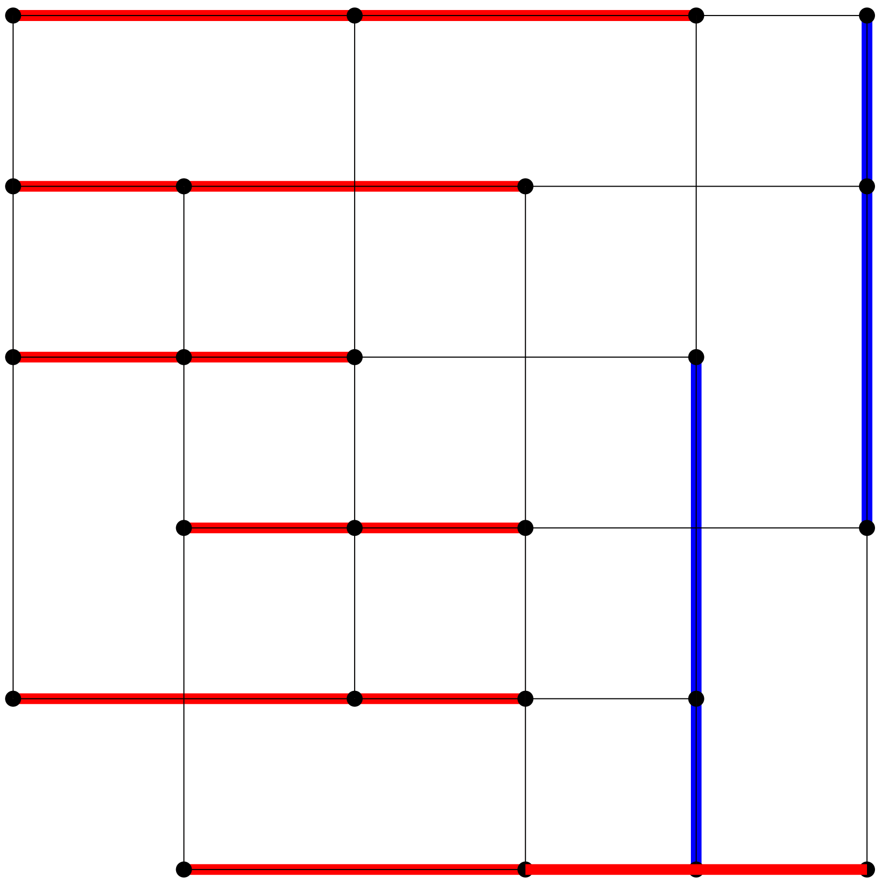


Figure 4.5: Modified line graph and S_3-decomposition using Strategy 3.

Chapter 5

Conclusion

We began this project with the aim of finding S_k-decompositions of bipartite graphs and answering the question, "Does an S_k-decomposition exist for a given bipartite graph?"

Through this project, we showed a proof by construction that complete bipartite graphs with n vertices on each partites set have an S_k-decomposition, if and only if k divides n^2 and $k \leq n$. We also showed that there is an S_k-decomposition for crown graphs with n vertices if and only if k divides $n(n-1)$ and $k \leq n-1$. We next showed that we can construct an S_k-decomposition for equipartite tripartite graphs with n vertices in each partite set, if and only if k divides $3n^2$ and $k \leq \frac{2}{3}n$. We showed that a complete tripartite graph $K_{p,q,r}$ has a S_3-decomposition if and only if $pq + pr + qr$ is divisible by 3, and $p, q, r \geq 2$ or if any two of the three partite sets have size divisible by 3.

The main obstacle faced in this project was dealing with the NP-Completeness of the decomposition problem. Often times we lose too much generality when constructing the test case and obtain results that are not useful for the general case of the graphs.

As noted in Chapter 4, it may be interesting to see if Strategy 2 is more efficient when the number of vertices in each partite set is more than 18. While Strategy 1 give results for $n < 39$ within a reasonable amount of time, the runtime of Strategy 1 grows exponentially and struggles to give results for $n \geq 42$. The results of Strategy 1 and 2 suggest that there is an S_3-decomposition for cyclic 4-regular bipartite graphs with one component when $n > 42$. It would be interesting to see if this is true for all n. There may be some additional structure not noted in Strategy 3 which may solve this conjecture.

Future work may include extending the results of Chapter 4 for S_3 -decomposition of cyclic r-regular bipartite graphs where $r \geq 5$. The primary reason why $r = 4$ was the focus of Chapter 4 was because, that case was the most restrictive but is the easiest to analyse. One suggestion as to how we may extend the case to $r = 5$ is to pick the first 4 elements of the generator set and then find a value x such we can offset the centers in partite set U without using the same center twice. Another suggestion is to check all five possible combinations of the generator set, and then find two sets of centers such that the two results do not use the same center twice.

Chapter 6

Appendix

6.1 Source Code for Strategy 1

In this section, we give the source code for the computer programme written to find the S_3-cover of partite set V for cyclic bipartite graphs (see Section 4.1.1). Minor details of the algorithm is included in the comment blocks of the source code.

6.1.1 The main wrapper program

```
import java.io.*;
import java.util.*;



public class genSolution {


  /**
   * @param args
   */
  static boolean outputLatex=true;
  //Generates Output as a Latex Table, worthwhile 3 hour
    investment

  public static void main(String[] args) {
    for (int a=2; a<40; a++) {
```

```
int size=a*3;
String fileName = "cyclic_size_"+size+".tex";
long start=System.nanoTime();

    try {
        // FileReader reads text files in the default encoding
.
        printWriterWrapper stream = new printWriterWrapper(
fileName,outputLatex);
        stream.print("\\begin{longtable}{|c|");
        for(int i=0; i<a;i++) {
            stream.print("c|");
        }
        stream.print("}\r\n");
        stream.println("\\hline");
        stream.print("\\begin{tabular}[c]{@{}c@{}}Generator
\\\\ Set\\end{tabular}& ");
        for (int i=1; i<a; i++) {
          stream.print("Star "+i+"\t& ");
        }
        stream.print("Star "+a+"\\\\ \r\n");
        stream.println("\\hline");
        stream.println("\\endfirsthead");
        stream.println("\\multicolumn{"+(a+1)+"}{c}%");
        stream.println("{\\tablename\\ \\thetable\\ -- \\
textit{Continued from previous page}} \\\\");
        stream.println("\\hline");
        stream.print("\\begin{tabular}[c]{@{}c@{}}Generator
\\\\ Set\\end{tabular}& ");
        for (int i=1; i<a; i++) {
          stream.print("Star "+i+"\t& ");
        }
        stream.print("Star "+a+"\\\\ \r\n");
        stream.println("\\hline");
        stream.println("\\endhead");
        stream.println("\\hline");
```

```
          stream.println("\\multicolumn{"+(a+1)+"}{c}%");
          stream.println("{\\tablename\\ \\thetable\\ -- \\
textit{Continued on next page}} \\\\");
          stream.println("\\endfoot");
          stream.println("\\hline");
          stream.println("\\caption{$S_3$-factor for Cyclic
Bipartite Graph $n="+size+"$}\\\\");
          stream.println("\\endlastfoot");

          // Always close files.
          double successRate=0;
          int success=0;
          int tries=0;

          System.out.println("Size: "+size);
          for (int diff=3;diff<=(size*3/4);diff++) {
            stream.flush();
            for(int i=1;i<=size-diff;i++) {
              for(int j=1;j<=size-diff;j++) {
                int k=diff-i-j;
                if (i+j>=diff) continue;
                if (k>size-diff) continue;
                int d1=i;
                int d2=i+j;
                int d3=i+j+k;
                if((i%3==0) && (j%3==0) && (k%3==0)) {
                      stream.println("\\cline{2-"+(a+1)+"}");
                  if((size/3)%3!=0) {
                    stream.print("$\\{0, "+d1+", "+d2+", "+d3+"
\\}$                                      & \\multicolumn
{"+a+"}{c|}{Three component graph, no decomposition}\\\\\r\n")
;
                  } else {
                    String details="$n="+size/3+"$ $D=\\{0, "+d1
/3+", "+d2/3+", "+d3/3+"\\}$";
```

```
                    stream.print("$\\{0, "+d1+", "+d2+", "+d3+"
\\}$                                          & \\multicolumn
{"+a+"}{c|}{Three component graph, see "+details+" }\\\\\r\n")
;
                  }
                      stream.println("\\cline{2-"+(a+1)+"}");

                  continue;
                };
                if((size%2==0) && (i%2==0) && (j%2==0) && (k
%2==0)) {
                      stream.println("\\cline{2-"+(a+1)+"}");
                  if((size/2)%3!=0) {
                    stream.print("$\\{0, "+d1+", "+d2+", "+d3+"
\\}$                                          & \\multicolumn
{"+a+"}{c|}{Two components graph, no decomposition}\\\\\r\n");
                  } else {
                    String details="$n="+size/2+"$ $D=\\{0, "+d1
/2+", "+d2/2+", "+d3/2+"\\}$";
                    stream.print("$\\{0, "+d1+", "+d2+", "+d3+"
\\}$                                          & \\multicolumn
{"+a+"}{c|}{Two component graph, see "+details+" }\\\\\r\n");
                  }
                      stream.println("\\cline{2-"+(a+1)+"}");
                  continue;
                }
                if((size%5==0) && (i%5==0) && (j%5==0) && (k
%5==0)) {
                      stream.println("\\cline{2-"+(a+1)+"}");
                  if((size/5)%3!=0) {
                    stream.print("$\\{0, "+d1+", "+d2+", "+d3+"
\\}$                                          & \\multicolumn
{"+a+"}{c|}{Five component graph, no decomposition}\\\\\r\n");
                  } else {
                    String details="$n="+size/5+"$ $D=\\{0, "+d1
/5+", "+d2/5+", "+d3/5+"\\}$";
```

```
                    stream.print("$\\{0, "+d1+", "+d2+", "+d3+"
\\}$                                                 & \\multicolumn
{"+a+"}{c|}{Five component graph, see "+details+" }\\\\\r\n");
                    }
                        stream.println("\\cline{2-"+(a+1)+"}");
                    continue;
                }
                tries++;
                cyclic c=new cyclic(0,i,i+j,i+j+k,size);
                cyclicList l=new cyclicList(c);
                List<Integer> solutions=new ArrayList<Integer>()
;
                solutions=l.generateSolution(6);
                if (solutions.size()<size/3) {
                    System.out.print(l.generateList(0)+"\t");
                  System.out.println("No solution for this
cyclic pattern");
                      stream.println("\\cline{2-"+(a+1)+"}");
                  stream.print("$\\{0, "+d1+", "+d2+", "+d3+"\\}
$                                                  & \\multicolumn{"+
a+"}{c|}{No solution using Strategy 1 }\\\\\r\n");
                      stream.println("\\cline{2-"+(a+1)+"}");
                } else {
                  success++;
                  latexTable(l,stream);
                }
              }
            }
          }

          successRate=(double) success / (double) tries * 100 ;
          System.out.println("Runs: "+ success + "/" +tries);
          System.out.println("Success Rate: "+ successRate);
      long runtime=System.nanoTime();
      double miliSec=(double) ((runtime-start)/1000000);
      double avgRun=(double) miliSec/tries;
```

```
            System.out.println("Runtime: "+ miliSec + "ms\t
  Average: "+avgRun);

            stream.println("\\end{longtable}");
            stream.close();
        }
        catch(IOException ex) {
           ex.printStackTrace();
        }



    }
}
public static void latexTable(cyclicList l, printWriterWrapper
  stream) throws IOException {
   int size=l.Seed.size;
   List<Integer> generator=l.generateList(0);
   stream.print("$\\{");
   int flag=0;
   for(int d:generator) {
     if (flag!=0) {
       stream.print(", ");
     }
     stream.print(d);
     flag=1;
   }
   stream.print("\\}$\t\t");


   for (int i=0; i<size; i++) {
     if (l.solOut.get(i) != null) {
       stream.print("& $(u_{"+i+"};");
       List<Integer> list= l.solOut.get(i);
       Collections.sort(list);
       flag=0;
```

```
            for(int v:list) {
                if (flag!=0) {
                    stream.print(",");
                }
                stream.print("v_{"+v+"}");
                flag=1;
            }
            stream.print(")$");
        }
    }
    stream.print("\\\\\r\n");
  }
}
```

genSolution.java

6.1.2 The solver

```
import java.util.*;



public class cyclicList {
  cyclic Seed;
  HashMap<Integer, List<Integer>> solOut = new HashMap<Integer,
    List<Integer>>();
  public cyclicList(cyclic s) {
    Seed=s;
  }
  public List<Integer> generateList(int offset) {
    List<Integer> r = new ArrayList<Integer>();
    int d=Seed.d1+offset>=Seed.size?Seed.d1+offset-Seed.size:Seed.
    d1+offset;
    r.add(d);
    d=Seed.d2+offset>=Seed.size?Seed.d2+offset-Seed.size:Seed.d2+
    offset;
    r.add(d);
    d=Seed.d3+offset>=Seed.size?Seed.d3+offset-Seed.size:Seed.d3+
    offset;
    r.add(d);
    d=Seed.d4+offset>=Seed.size?Seed.d4+offset-Seed.size:Seed.d4+
    offset;
    r.add(d);
    Collections.sort(r);
    return r;
  }



  public List<Integer> generateSolution(int algorithm){
    List<Integer> solutions=new ArrayList<Integer>();

  if(algorithm==6) {
    /* brute force, checks for entire search space*/
    int flag=0;
```

```
 int runTime=0;
 HashMap<Integer, Integer> counter = new HashMap<Integer,
Integer>();
 counter.put(0, 0);
 for (int i=1; i<Seed.size/3; i++) {
   counter.put(i, 1);
 }
 while (flag==0) {
   runTime++;
   if(runTime>1000000000) {
     /* always a good practice to make sure we don't end in an
infinite loop */
     System.out.println("runtime exceeded");
     flag=1;
   }
   int sum=0;
   int partialFailed=0;
   List<Integer> test = new ArrayList<Integer>();
   test.add(0);
   for (int i=1; i<Seed.size/3; i++) {
     sum+=counter.get(i);
     test.add(sum);
     if (partialCheckSolution(test)==false) {
       partialFailed=i;
       i=Seed.size;
     }
   }
   if(partialFailed>0) {
     for(int i=partialFailed+1;i<Seed.size/3;i++) {
       counter.put(i,1);
     }
     for(int i=partialFailed;i>=1;i--) {
       int val=counter.get(i);
       sum=0;
       for (int j=1; j<Seed.size/3; j++) {
         sum+=counter.get(j);
```

```
        }
        if(sum<Seed.size) {
          val++;
          counter.put(i,val);
          i=0;
          continue;
        } else {
          if(i==1) {flag=1;}
          counter.put(i,1);
        }
      }
    } else if (checkSolution(test)) {
      solutions=test;
      return solutions;
    } else {
    for(int i=Seed.size/3-1;i>=1;i--) {
      int val=counter.get(i);
      sum=0;
      for (int j=1; j<Seed.size/3; j++) {
        sum+=counter.get(j);
      }
      if(sum<Seed.size) {
        val++;
        counter.put(i,val);
        i=0;
        continue;
      } else {
        if(i==1) {flag=1;}
        counter.put(i,1);
      }
    }

      sum=0;
      for (int i=1; i<Seed.size/3; i++) {
        sum+=counter.get(i);
        if(sum>Seed.size) flag=1;
```

```
        }

      }



    }
  }



  return solutions;
  }

  public boolean partialCheckSolution(List<Integer> test) {
    int size=test.size();
    HashMap<Integer, Integer> check = new HashMap<Integer, Integer
    >();
    for(int offset:test) {
      for(int val:generateList(offset)) {
        check.put(val,1);
      }
    }
    /* let k = n/3 - size of partial solution
     * if n- edge covered by partial solutions > 3*k then clearly
    adding
     * k additional solutions not give us a solution
     * this check speeds things up by a factor of 3
     */
    if(check.size()<3*size) return false;
    return true;
  }

  public HashMap<Integer, List<Integer>> getSolution(HashMap<
    Integer, List<Integer>> candidates, List<Integer> unsolved,
    HashMap<Integer, List<Integer>> out) {
    HashMap<Integer, Integer> sizeOfCandidates = new HashMap<
    Integer, Integer>();
```

```
HashMap<Integer, List<Integer>> sizeOfMissing = new HashMap<
Integer, List<Integer>>();

for(int i=0; i<5; i++) {
  sizeOfMissing.put(i,new ArrayList<Integer>());
}
for(int c1=0; c1<Seed.size; c1++) {
  if (candidates.get(c1)!=null) {
    List<Integer> hold = candidates.get(c1);
    if (hold.size()>1) {
      int sizeMiss=hold.size();
      List<Integer> tempMiss=sizeOfMissing.get(sizeMiss);
      tempMiss.add(c1);
      sizeOfMissing.put(sizeMiss,tempMiss);
      for(int c2:hold) {
        int size=out.get(c2).size();
        sizeOfCandidates.put(c2,size);
      }
    } else {
      candidates.remove(c1);
    }
  }
}

List<Integer> missing=new ArrayList<Integer>();
for(int i=0; i<4; i++) {
  missing.addAll(sizeOfMissing.get(i));
}
while(missing.isEmpty()==false){
  int c1=missing.get(0);
  List<Integer> list=candidates.get(c1);
  int choice=-1;
  int lowSeen=999;
  for(int c2:list) {
    if (lowSeen>sizeOfCandidates.get(c2)){
      choice=c2;
```

```
        lowSeen=sizeOfCandidates.get(c2);
      }
    }
    if (choice!=-1) {
      List<Integer> temp=out.get(choice);
      temp.add(c1);
      int temp2=sizeOfCandidates.get(choice);
      temp2++;
      sizeOfCandidates.put(choice,temp2);
      candidates.remove(c1);
      if(temp2==3) {
        for(int i=0; i<Seed.size; i++) {
          if (candidates.get(i)!=null) {
            List<Integer> hold = candidates.get(i);
            if (hold.contains(choice))  hold.remove(hold.indexOf
(choice));
            candidates.put(i, hold);
          }
        }
      }
    } else {
      // This shouldn't happen, since the previous step
guarantees that the edge belongs to
      // at least one center u_i, but if this does happen then
clearly c(x) is not a valid center function
      System.out.println("No Solution");
    }
    missing.clear();
    for(int i=0; i<5; i++) {
      sizeOfMissing.put(i,new ArrayList<Integer>());
    }
    for(int c3=0; c3<Seed.size; c3++) {
      if (candidates.get(c3)!=null) {
        List<Integer> hold = candidates.get(c3);
        int sizeMiss=hold.size();
        List<Integer> tempMiss=sizeOfMissing.get(sizeMiss);
```

```
          tempMiss.add(c3);
          sizeOfMissing.put(sizeMiss,tempMiss);
        }
      }
      for(int i=0; i<4; i++) {
        missing.addAll(sizeOfMissing.get(i));
      }



    }
    return out;
  }

  public boolean checkSolution(List<Integer> solutions) {
    HashMap<Integer, Integer> check = new HashMap<Integer, Integer
   >();
    for(int c1=0; c1<Seed.size; c1++) {
      check.put(c1,0);
    }
    for(int offset:solutions) {
      for(int val:generateList(offset)) {
        check.put(val,(check.get(val)+1));
      }
    }
    for(int c1=0; c1<Seed.size; c1++) {
      //If edge {x,y} does not belong to a center c(x), then
   condition 3 is violated
      if(check.get(c1)==0) return false;
    }

    for(int offset:solutions) {
      int count=0;
      for(int val:generateList(offset)) {
        if(check.get(val)==1) count++;
      }
      if (count==4) return false;
```

```
    }

    // Just because condition 3 is met, does not mean that c(x) is
     a center function,
    // we need to make sure that c(x) is a valid center function;
    return doubleCheckSolution(solutions);
  }

  public void printSolution(List<Integer> solutions) {
    System.out.println("Solution: " + solutions + "\tOutput: "+
   solOut);
  }



public boolean doubleCheckSolution(List<Integer> solutions) {
  HashMap<Integer, List<Integer>> check = new HashMap<Integer,
    List<Integer>>();
  HashMap<Integer, List<Integer>> candidates = new HashMap<Integer
    , List<Integer>>();
  HashMap<Integer, List<Integer>> out = new HashMap<Integer, List<
    Integer>>();
  HashMap<Integer, List<Integer>> list = new HashMap<Integer, List
   <Integer>>();

  /* We make sure that c(x) is a valid center function */
  List<Integer> missing=new ArrayList<Integer>();
  for(int c1=0; c1<Seed.size; c1++) {
    List<Integer> temp = new ArrayList<Integer>();
    List<Integer> temp2 = new ArrayList<Integer>();

    check.put(c1, temp);
    candidates.put(c1,temp2);
  }
  for(int offset:solutions) {
    List<Integer> temp2 = new ArrayList<Integer>();
```

```
    out.put(offset,temp2);
    for(int val:generateList(offset)) {
      List<Integer> temp=check.get(val);
      temp.add(offset);
      check.put(val,temp);
    }
  }
  for(int c1=0; c1<Seed.size; c1++) {
    List<Integer> temp=check.get(c1);
    if(temp.size()==1) {
      List<Integer> temp2=out.get(temp.get(0));
      temp2.add(c1);
      out.put(temp.get(0),temp2);
      } else if (temp.size()==0) {
      // the v_c1 is not adjacent to a center, therefore c(x) is
   not a valid center function
      // This should not happen since it is guaranteed by the
   previous step that v_c1 is adjacent to a center
      System.out.println("Invalid Solution for "+c1);
      return false;
    }
  }
  for(int c1=0; c1<Seed.size; c1++) {
    List<Integer> temp=check.get(c1);
    if(temp.size()>1) {
      for(int test:temp) {
        List<Integer> temp2=out.get(test);
        if (temp2.size()<3) {
          List<Integer> hold=candidates.get(c1);
          hold.add(test);
          candidates.put(c1,hold);
        }
      }
      List<Integer> hold=candidates.get(c1);
      if(hold.size()==1) {
        List<Integer> temp2=out.get(hold.get(0));
```

```
        temp2.add(c1);
        out.put(hold.get(0),temp2);
        candidates.remove(c1);
      } else {
        missing.add(c1);
        list.put(c1, hold);
      }
    }




  }
  Collections.sort(solutions);
  out = getSolution(list,missing,out);

  for(int c3=0; c3<Seed.size; c3++) {
    if (out.get(c3)!=null) {
      if(out.get(c3).size()<3) {
        // not every u_c3 has size 3, therefore c(x) is not a
   valid function
        return false;
      }
    }
  }
  solOut=out;
  return true;

}


}
```

cyclicList.java

6.1.3 Supporting JAVA classes

```
import java.lang.*;
import java.util.*;
public class cyclic {

  public int d1,d2,d3,d4;
  public int size;

  public cyclic(int d_1, int d_2, int d_3, int d_4, int s) {
    @SuppressWarnings("unchecked")
    List<Integer> test=new ArrayList<Integer>();
    test.add(d_1);
    test.add(d_2);
    test.add(d_3);
    test.add(d_4);
    Collections.sort(test);
    d1=test.get(0);
    d2=test.get(1);
    d3=test.get(2);
    d4=test.get(3);
    size=s;
  }

}
```

cyclic.java

6.2 S_3-cover of partite set V

In this section we give the results of the output of our computer programme for cyclic bipartite graphs of size $n \leq 18$ (see Section 4.1.1). The following tables gives us the copies of S_3 with centers in U such that each vertex in V is used exactly once.

6.2.1 S_3-cover of partite set V for $n = 6$

Generator Set	Star 1	Star 2
$\{0,1,2,3\}$	$(u_0; v_0, v_1, v_2)$	$(u_2; v_3, v_4, v_5)$
$\{0,1,2,4\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_3, v_5)$
$\{0,1,3,4\}$	$(u_0; v_0, v_1, v_3)$	$(u_1; v_2, v_4, v_5)$
$\{0,2,3,4\}$	$(u_0; v_0, v_2, v_3)$	$(u_1; v_1, v_4, v_5)$

Table 6.1: S_3-cover of Partite Set V for $n = 6$

6.2.2 S_3-cover of partite set V for $n = 9$

Generator Set	Star 1	Star 2	Star 3
$\{0,1,2,3\}$	$(u_0; v_0, v_1, v_2)$	$(u_2; v_3, v_4, v_5)$	$(u_5; v_6, v_7, v_8)$
$\{0,1,2,4\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_3, v_5)$	$(u_6; v_6, v_7, v_8)$
$\{0,1,3,4\}$	$(u_0; v_0, v_1, v_3)$	$(u_2; v_2, v_5, v_6)$	$(u_4; v_4, v_7, v_8)$
$\{0,2,3,4\}$	$(u_0; v_0, v_2, v_3)$	$(u_1; v_1, v_4, v_5)$	$(u_4; v_6, v_7, v_8)$
$\{0,1,2,5\}$	$(u_0; v_0, v_1, v_5)$	$(u_2; v_2, v_3, v_4)$	$(u_6; v_6, v_7, v_8)$
$\{0,1,3,5\}$	$(u_0; v_0, v_1, v_5)$	$(u_1; v_2, v_4, v_6)$	$(u_7; v_3, v_7, v_8)$
$\{0,1,4,5\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_5, v_6)$	$(u_3; v_3, v_7, v_8)$
$\{0,2,3,5\}$	$(u_0; v_0, v_2, v_3)$	$(u_1; v_1, v_4, v_6)$	$(u_5; v_5, v_7, v_8)$
$\{0,2,4,5\}$	$(u_0; v_0, v_2, v_4)$	$(u_1; v_1, v_3, v_6)$	$(u_3; v_5, v_7, v_8)$

Table 6.2 – *Continued on next page*

Table 6.2 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3
$\{0,3,4,5\}$	$(u_0; v_0, v_3, v_4)$	$(u_2; v_2, v_6, v_7)$	$(u_5; v_1, v_5, v_8)$
$\{0,1,3,6\}$	$(u_0; v_0, v_1, v_6)$	$(u_1; v_2, v_4, v_7)$	$(u_2; v_3, v_5, v_8)$
$\{0,1,4,6\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_5, v_7)$	$(u_2; v_3, v_6, v_8)$
$\{0,2,3,6\}$	$(u_0; v_0, v_2, v_6)$	$(u_1; v_1, v_3, v_7)$	$(u_2; v_4, v_5, v_8)$
$\{0,2,4,6\}$	$(u_0; v_0, v_2, v_4)$	$(u_1; v_3, v_5, v_7)$	$(u_4; v_1, v_6, v_8)$
$\{0,2,5,6\}$	$(u_0; v_0, v_2, v_5)$	$(u_1; v_1, v_3, v_6)$	$(u_2; v_4, v_7, v_8)$
$\{0,3,4,6\}$	$(u_0; v_0, v_3, v_4)$	$(u_1; v_1, v_5, v_7)$	$(u_2; v_2, v_6, v_8)$
$\{0,3,5,6\}$	$(u_0; v_0, v_3, v_5)$	$(u_1; v_1, v_4, v_6)$	$(u_2; v_2, v_7, v_8)$

Table 6.2: S_3-cover of Partite Set V for $n = 9$

6.2.3 S_3-cover of partite set V for $n = 12$

Generator Set	Star 1	Star 2	Star 3	Star 4
$\{0,1,2,3\}$	$(u_0; v_0, v_1, v_3)$	$(u_2; v_2, v_4, v_5)$	$(u_5; v_6, v_7, v_8)$	$(u_8; v_9, v_{10}, v_{11})$
$\{0,1,2,4\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_3, v_5)$	$(u_6; v_6, v_7, v_{10})$	$(u_7; v_8, v_9, v_{11})$
$\{0,1,3,4\}$	$(u_0; v_0, v_3, v_4)$	$(u_1; v_1, v_2, v_5)$	$(u_5; v_6, v_8, v_9)$	$(u_7; v_7, v_{10}, v_{11})$
$\{0,2,3,4\}$	$(u_0; v_0, v_2, v_3)$	$(u_1; v_1, v_4, v_5)$	$(u_4; v_6, v_7, v_8)$	$(u_7; v_9, v_{10}, v_{11})$
$\{0,1,2,5\}$	$(u_0; v_0, v_1, v_5)$	$(u_2; v_2, v_3, v_4)$	$(u_6; v_6, v_7, v_{11})$	$(u_8; v_8, v_9, v_{10})$
$\{0,1,3,5\}$	$(u_0; v_0, v_1, v_3)$	$(u_1; v_2, v_4, v_6)$	$(u_5; v_5, v_8, v_{10})$	$(u_6; v_7, v_9, v_{11})$
$\{0,1,4,5\}$	$(u_0; v_0, v_1, v_4)$	$(u_2; v_2, v_3, v_6)$	$(u_4; v_5, v_8, v_9)$	$(u_6; v_7, v_{10}, v_{11})$
$\{0,2,3,5\}$	$(u_0; v_0, v_2, v_5)$	$(u_1; v_1, v_3, v_4)$	$(u_4; v_6, v_7, v_9)$	$(u_8; v_8, v_{10}, v_{11})$
$\{0,2,4,5\}$	$(u_0; v_0, v_2, v_4)$	$(u_1; v_1, v_3, v_5)$	$(u_5; v_7, v_9, v_{10})$	$(u_6; v_6, v_8, v_{11})$
$\{0,3,4,5\}$	$(u_0; v_0, v_3, v_4)$	$(u_2; v_2, v_6, v_7)$	$(u_5; v_5, v_9, v_{10})$	$(u_8; v_1, v_8, v_{11})$
$\{0,1,2,6\}$	$(u_0; v_0, v_1, v_6)$	$(u_2; v_3, v_4, v_8)$	$(u_5; v_5, v_7, v_{11})$	$(u_8; v_2, v_9, v_{10})$
$\{0,1,3,6\}$	$(u_0; v_0, v_3, v_6)$	$(u_1; v_1, v_2, v_4)$	$(u_4; v_5, v_7, v_{10})$	$(u_8; v_8, v_9, v_{11})$

Table 6.3 – *Continued on next page*

Table 6.3 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4
$\{0,1,4,6\}$	$(u_0;v_0,v_4,v_6)$	$(u_1;v_1,v_2,v_5)$	$(u_7;v_7,v_8,v_{11})$	$(u_9;v_3,v_9,v_{10})$
$\{0,1,5,6\}$	$(u_0;v_0,v_1,v_5)$	$(u_1;v_2,v_6,v_7)$	$(u_3;v_3,v_8,v_9)$	$(u_{10};v_4,v_{10},v_{11})$
$\{0,2,3,6\}$	$(u_0;v_0,v_2,v_3)$	$(u_1;v_1,v_4,v_7)$	$(u_3;v_5,v_6,v_9)$	$(u_8;v_8,v_{10},v_{11})$
$\{0,2,4,6\}$	Two-component graph see $n=6$ and $D=\{0,1,2,3\}$			
$\{0,2,5,6\}$	$(u_0;v_0,v_2,v_5)$	$(u_1;v_1,v_3,v_7)$	$(u_4;v_4,v_9,v_{10})$	$(u_6;v_6,v_8,v_{11})$
$\{0,3,4,6\}$	$(u_0;v_0,v_3,v_6)$	$(u_1;v_1,v_5,v_7)$	$(u_5;v_8,v_9,v_{11})$	$(u_{10};v_2,v_4,v_{10})$
$\{0,3,5,6\}$	$(u_0;v_0,v_3,v_5)$	$(u_1;v_1,v_4,v_6)$	$(u_4;v_7,v_9,v_{10})$	$(u_8;v_2,v_8,v_{11})$
$\{0,4,5,6\}$	$(u_0;v_0,v_4,v_6)$	$(u_2;v_2,v_7,v_8)$	$(u_5;v_5,v_{10},v_{11})$	$(u_9;v_1,v_3,v_9)$
$\{0,1,2,7\}$	$(u_0;v_0,v_1,v_7)$	$(u_1;v_2,v_3,v_8)$	$(u_4;v_4,v_5,v_6)$	$(u_9;v_9,v_{10},v_{11})$
$\{0,1,3,7\}$	$(u_0;v_0,v_1,v_3)$	$(u_1;v_2,v_4,v_8)$	$(u_6;v_6,v_7,v_9)$	$(u_{10};v_5,v_{10},v_{11})$
$\{0,1,4,7\}$	$(u_0;v_0,v_4,v_7)$	$(u_1;v_1,v_2,v_8)$	$(u_2;v_3,v_6,v_9)$	$(u_{10};v_5,v_{10},v_{11})$
$\{0,1,5,7\}$	$(u_0;v_0,v_1,v_7)$	$(u_1;v_2,v_6,v_8)$	$(u_3;v_3,v_4,v_{10})$	$(u_4;v_5,v_9,v_{11})$
$\{0,1,6,7\}$	$(u_0;v_0,v_1,v_6)$	$(u_1;v_2,v_7,v_8)$	$(u_3;v_3,v_4,v_9)$	$(u_4;v_5,v_{10},v_{11})$
$\{0,2,3,7\}$	$(u_0;v_0,v_2,v_7)$	$(u_1;v_1,v_3,v_8)$	$(u_3;v_5,v_6,v_{10})$	$(u_9;v_4,v_9,v_{11})$
$\{0,2,4,7\}$	$(u_0;v_0,v_2,v_4)$	$(u_1;v_1,v_3,v_5)$	$(u_6;v_6,v_8,v_{10})$	$(u_7;v_7,v_9,v_{11})$
$\{0,2,5,7\}$	$(u_0;v_0,v_2,v_7)$	$(u_1;v_1,v_3,v_6)$	$(u_3;v_5,v_8,v_{10})$	$(u_4;v_4,v_9,v_{11})$
$\{0,2,6,7\}$	$(u_0;v_0,v_2,v_7)$	$(u_1;v_1,v_3,v_8)$	$(u_3;v_5,v_9,v_{10})$	$(u_4;v_4,v_6,v_{11})$
$\{0,3,4,7\}$	$(u_0;v_0,v_3,v_4)$	$(u_1;v_1,v_5,v_8)$	$(u_2;v_2,v_6,v_9)$	$(u_7;v_7,v_{10},v_{11})$
$\{0,3,5,7\}$	$(u_0;v_0,v_3,v_5)$	$(u_1;v_1,v_6,v_8)$	$(u_4;v_4,v_9,v_{11})$	$(u_7;v_2,v_7,v_{10})$
$\{0,3,6,7\}$	$(u_0;v_0,v_3,v_6)$	$(u_1;v_1,v_4,v_8)$	$(u_2;v_2,v_5,v_9)$	$(u_4;v_7,v_{10},v_{11})$
$\{0,4,5,7\}$	$(u_0;v_0,v_4,v_7)$	$(u_1;v_1,v_5,v_6)$	$(u_4;v_8,v_9,v_{11})$	$(u_{10};v_2,v_3,v_{10})$
$\{0,4,6,7\}$	$(u_0;v_0,v_4,v_6)$	$(u_1;v_1,v_5,v_8)$	$(u_3;v_3,v_9,v_{10})$	$(u_7;v_2,v_7,v_{11})$
$\{0,5,6,7\}$	$(u_0;v_0,v_5,v_6)$	$(u_1;v_1,v_7,v_8)$	$(u_4;v_4,v_{10},v_{11})$	$(u_9;v_2,v_3,v_9)$
$\{0,1,4,8\}$	$(u_0;v_0,v_1,v_8)$	$(u_1;v_2,v_5,v_9)$	$(u_2;v_3,v_6,v_{10})$	$(u_3;v_4,v_7,v_{11})$
$\{0,1,5,8\}$	$(u_0;v_0,v_1,v_5)$	$(u_1;v_2,v_6,v_9)$	$(u_2;v_3,v_7,v_{10})$	$(u_3;v_4,v_8,v_{11})$
$\{0,2,4,8\}$	Two-component graph see $n=6$ and $D=\{0,1,2,4\}$			

Table 6.3 – *Continued on next page*

Table 6.3 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4
$\{0,2,5,8\}$	$(u_0; v_0, v_2, v_5)$	$(u_1; v_1, v_6, v_9)$	$(u_2; v_4, v_7, v_{10})$	$(u_3; v_3, v_8, v_{11})$
$\{0,2,6,8\}$	Two-component graph see $n = 6$ and $D = \{0,1,3,4\}$			
$\{0,3,4,8\}$	$(u_0; v_0, v_3, v_8)$	$(u_1; v_1, v_4, v_9)$	$(u_2; v_2, v_5, v_{10})$	$(u_3; v_6, v_7, v_{11})$
$\{0,3,5,8\}$	$(u_0; v_0, v_3, v_5)$	$(u_1; v_1, v_4, v_9)$	$(u_2; v_2, v_7, v_{10})$	$(u_3; v_6, v_8, v_{11})$
$\{0,3,6,8\}$	$(u_0; v_0, v_3, v_8)$	$(u_1; v_1, v_4, v_7)$	$(u_2; v_2, v_5, v_{10})$	$(u_3; v_6, v_9, v_{11})$
$\{0,3,7,8\}$	$(u_0; v_0, v_3, v_7)$	$(u_1; v_1, v_4, v_8)$	$(u_2; v_2, v_5, v_9)$	$(u_3; v_6, v_{10}, v_{11})$
$\{0,4,5,8\}$	$(u_0; v_0, v_4, v_5)$	$(u_1; v_1, v_6, v_9)$	$(u_2; v_2, v_7, v_{10})$	$(u_3; v_3, v_8, v_{11})$
$\{0,4,6,8\}$	Two-component graph see $n = 6$ and $D = \{0,2,3,4\}$			
$\{0,4,7,8\}$	$(u_0; v_0, v_4, v_7)$	$(u_1; v_1, v_5, v_8)$	$(u_2; v_2, v_6, v_9)$	$(u_3; v_3, v_{10}, v_{11})$
$\{0,3,6,9\}$	Three-component graphs, no decomposition			

Table 6.3: S_3-cover of Partite Set V for $n = 12$

6.2.4 S_3-cover of partite set V for $n = 15$

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5
$\{0,1,2,3\}$	$(u_0; v_0, v_1, v_3)$	$(u_2; v_2, v_4, v_5)$	$(u_5; v_6, v_7, v_8)$	$(u_8; v_9, v_{10}, v_{11})$	$(u_{11}; v_{12}, v_{13}, v_{14})$
$\{0,1,2,4\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_3, v_5)$	$(u_6; v_6, v_7, v_{10})$	$(u_7; v_8, v_9, v_{11})$	$(u_{12}; v_{12}, v_{13}, v_{14})$
$\{0,1,3,4\}$	$(u_0; v_0, v_1, v_3)$	$(u_1; v_2, v_4, v_5)$	$(u_6; v_6, v_7, v_9)$	$(u_8; v_8, v_{11}, v_{12})$	$(u_{10}; v_{10}, v_{13}, v_{14})$
$\{0,2,3,4\}$	$(u_0; v_0, v_2, v_3)$	$(u_1; v_1, v_4, v_5)$	$(u_4; v_6, v_7, v_8)$	$(u_7; v_9, v_{10}, v_{11})$	$(u_{10}; v_{12}, v_{13}, v_{14})$
$\{0,1,2,5\}$	$(u_0; v_0, v_1, v_5)$	$(u_2; v_2, v_3, v_4)$	$(u_6; v_6, v_7, v_{11})$	$(u_8; v_8, v_9, v_{10})$	$(u_{12}; v_{12}, v_{13}, v_{14})$
$\{0,1,3,5\}$	$(u_0; v_0, v_1, v_3)$	$(u_1; v_2, v_4, v_6)$	$(u_4; v_5, v_7, v_9)$	$(u_8; v_8, v_{11}, v_{13})$	$(u_9; v_{10}, v_{12}, v_{14})$
$\{0,1,4,5\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_5, v_6)$	$(u_3; v_3, v_7, v_8)$	$(u_8; v_9, v_{12}, v_{13})$	$(u_{10}; v_{10}, v_{11}, v_{14})$
$\{0,2,3,5\}$	$(u_0; v_0, v_2, v_5)$	$(u_1; v_1, v_3, v_6)$	$(u_4; v_4, v_7, v_9)$	$(u_8; v_8, v_{10}, v_{13})$	$(u_9; v_{11}, v_{12}, v_{14})$
$\{0,2,4,5\}$	$(u_0; v_0, v_2, v_4)$	$(u_1; v_1, v_5, v_6)$	$(u_3; v_3, v_7, v_8)$	$(u_8; v_{10}, v_{12}, v_{13})$	$(u_9; v_9, v_{11}, v_{14})$
$\{0,3,4,5\}$	$(u_0; v_0, v_3, v_4)$	$(u_2; v_2, v_6, v_7)$	$(u_5; v_5, v_9, v_{10})$	$(u_8; v_8, v_{12}, v_{13})$	$(u_{11}; v_1, v_{11}, v_{14})$
$\{0,1,2,6\}$	$(u_0; v_0, v_2, v_6)$	$(u_1; v_1, v_3, v_7)$	$(u_3; v_4, v_5, v_9)$	$(u_8; v_8, v_{10}, v_{14})$	$(u_{11}; v_{11}, v_{12}, v_{13})$
$\{0,1,3,6\}$	$(u_0; v_0, v_3, v_6)$	$(u_1; v_2, v_4, v_7)$	$(u_5; v_5, v_8, v_{11})$	$(u_9; v_9, v_{10}, v_{12})$	$(u_{13}; v_1, v_{13}, v_{14})$

Table 6.4 – *Continued on next page*

Table 6.4 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5
$\{0,1,4,6\}$	$(u_0;v_0,v_4,v_6)$	$(u_1;v_2,v_5,v_7)$	$(u_8;v_8,v_9,v_{14})$	$(u_{10};v_1,v_{10},v_{11})$	$(u_{12};v_3,v_{12},v_{13})$
$\{0,1,5,6\}$	$(u_0;v_0,v_1,v_5)$	$(u_2;v_2,v_3,v_7)$	$(u_4;v_4,v_9,v_{10})$	$(u_6;v_6,v_{11},v_{12})$	$(u_8;v_8,v_{13},v_{14})$
$\{0,2,3,6\}$	$(u_0;v_0,v_2,v_3)$	$(u_1;v_1,v_4,v_7)$	$(u_3;v_5,v_6,v_9)$	$(u_8;v_8,v_{11},v_{14})$	$(u_{10};v_{10},v_{12},v_{13})$
$\{0,2,4,6\}$	$(u_0;v_0,v_2,v_4)$	$(u_1;v_3,v_5,v_7)$	$(u_4;v_6,v_8,v_{10})$	$(u_7;v_9,v_{11},v_{13})$	$(u_{10};v_1,v_{12},v_{14})$
$\{0,2,5,6\}$	$(u_0;v_0,v_2,v_5)$	$(u_1;v_1,v_3,v_7)$	$(u_4;v_4,v_6,v_9)$	$(u_6;v_8,v_{11},v_{12})$	$(u_8;v_{10},v_{13},v_{14})$
$\{0,3,4,6\}$	$(u_0;v_0,v_3,v_6)$	$(u_1;v_1,v_4,v_5)$	$(u_4;v_7,v_8,v_{10})$	$(u_9;v_9,v_{12},v_{13})$	$(u_{11};v_2,v_{11},v_{14})$
$\{0,3,5,6\}$	$(u_0;v_0,v_3,v_5)$	$(u_1;v_1,v_4,v_7)$	$(u_3;v_6,v_8,v_9)$	$(u_7;v_{10},v_{12},v_{13})$	$(u_{11};v_2,v_{11},v_{14})$
$\{0,4,5,6\}$	$(u_0;v_0,v_4,v_5)$	$(u_1;v_1,v_6,v_7)$	$(u_5;v_9,v_{10},v_{11})$	$(u_8;v_8,v_{13},v_{14})$	$(u_{12};v_2,v_3,v_{12})$
$\{0,1,2,7\}$	$(u_0;v_0,v_2,v_7)$	$(u_1;v_1,v_3,v_8)$	$(u_4;v_4,v_5,v_6)$	$(u_9;v_9,v_{10},v_{11})$	$(u_{12};v_{12},v_{13},v_{14})$
$\{0,1,3,7\}$	No solution using Strategy 1				
$\{0,1,4,7\}$	$(u_0;v_0,v_4,v_7)$	$(u_1;v_2,v_5,v_8)$	$(u_2;v_3,v_6,v_9)$	$(u_{10};v_{10},v_{11},v_{14})$	$(u_{12};v_1,v_{12},v_{13})$
$\{0,1,5,7\}$	$(u_0;v_0,v_1,v_7)$	$(u_1;v_2,v_6,v_8)$	$(u_4;v_4,v_9,v_{11})$	$(u_5;v_5,v_{10},v_{12})$	$(u_{13};v_3,v_{13},v_{14})$
$\{0,1,6,7\}$	$(u_0;v_0,v_1,v_6)$	$(u_1;v_2,v_7,v_8)$	$(u_3;v_3,v_9,v_{10})$	$(u_5;v_5,v_{11},v_{12})$	$(u_{13};v_4,v_{13},v_{14})$

Table 6.4 – *Continued on next page*

Table 6.4 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5
$\{0,2,3,7\}$	$(u_0; v_0, v_2, v_7)$	$(u_1; v_1, v_4, v_8)$	$(u_3; v_5, v_6, v_{10})$	$(u_9; v_9, v_{11}, v_{12})$	$(u_{11}; v_3, v_{13}, v_{14})$
$\{0,2,4,7\}$	$(u_0; v_0, v_2, v_7)$	$(u_1; v_3, v_5, v_8)$	$(u_2; v_4, v_6, v_9)$	$(u_9; v_1, v_{11}, v_{13})$	$(u_{10}; v_{10}, v_{12}, v_{14})$
$\{0,2,5,7\}$	$(u_0; v_0, v_2, v_5)$	$(u_1; v_1, v_3, v_6)$	$(u_4; v_4, v_9, v_{11})$	$(u_7; v_7, v_{12}, v_{14})$	$(u_8; v_8, v_{10}, v_{13})$
$\{0,2,6,7\}$	$(u_0; v_0, v_2, v_6)$	$(u_1; v_1, v_7, v_8)$	$(u_3; v_5, v_9, v_{10})$	$(u_{11}; v_3, v_{11}, v_{13})$	$(u_{12}; v_4, v_{12}, v_{14})$
$\{0,3,4,7\}$	$(u_0; v_0, v_3, v_7)$	$(u_1; v_1, v_4, v_8)$	$(u_2; v_2, v_5, v_6)$	$(u_7; v_{10}, v_{11}, v_{14})$	$(u_9; v_9, v_{12}, v_{13})$
$\{0,3,5,7\}$	$(u_0; v_0, v_3, v_5)$	$(u_1; v_1, v_4, v_8)$	$(u_2; v_2, v_7, v_9)$	$(u_6; v_6, v_{11}, v_{13})$	$(u_7; v_{10}, v_{12}, v_{14})$
$\{0,3,6,7\}$	$(u_0; v_0, v_3, v_6)$	$(u_1; v_1, v_4, v_7)$	$(u_2; v_2, v_5, v_9)$	$(u_5; v_8, v_{11}, v_{12})$	$(u_7; v_{10}, v_{13}, v_{14})$
$\{0,4,5,7\}$	$(u_0; v_0, v_4, v_5)$	$(u_1; v_1, v_6, v_8)$	$(u_5; v_9, v_{10}, v_{12})$	$(u_7; v_7, v_{11}, v_{14})$	$(u_{13}; v_2, v_3, v_{13})$
$\{0,4,6,7\}$	No solution using Strategy 1				
$\{0,5,6,7\}$	$(u_0; v_0, v_5, v_6)$	$(u_1; v_1, v_7, v_8)$	$(u_4; v_4, v_9, v_{10})$	$(u_7; v_{12}, v_{13}, v_{14})$	$(u_{11}; v_2, v_3, v_{11})$
$\{0,1,2,8\}$	$(u_0; v_0, v_1, v_8)$	$(u_2; v_2, v_3, v_4)$	$(u_5; v_5, v_6, v_7)$	$(u_9; v_9, v_{10}, v_{11})$	$(u_{12}; v_{12}, v_{13}, v_{14})$
$\{0,1,3,8\}$	$(u_0; v_0, v_3, v_8)$	$(u_1; v_1, v_2, v_4)$	$(u_4; v_5, v_7, v_{12})$	$(u_6; v_6, v_9, v_{14})$	$(u_{10}; v_{10}, v_{11}, v_{13})$
$\{0,1,4,8\}$	$(u_0; v_0, v_4, v_8)$	$(u_1; v_1, v_2, v_9)$	$(u_3; v_3, v_7, v_{11})$	$(u_6; v_6, v_{10}, v_{14})$	$(u_{12}; v_5, v_{12}, v_{13})$

Table 6.4 – *Continued on next page*

Table 6.4 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5
$\{0,1,5,8\}$	$(u_0; v_0, v_5, v_8)$	$(u_1; v_1, v_2, v_9)$	$(u_2; v_3, v_7, v_{10})$	$(u_{11}; v_4, v_{11}, v_{12})$	$(u_{13}; v_6, v_{13}, v_{14})$
$\{0,1,6,8\}$	$(u_0; v_0, v_1, v_8)$	$(u_1; v_2, v_7, v_9)$	$(u_3; v_3, v_4, v_{11})$	$(u_4; v_5, v_{10}, v_{12})$	$(u_{13}; v_6, v_{13}, v_{14})$
$\{0,1,7,8\}$	$(u_0; v_0, v_1, v_7)$	$(u_1; v_2, v_8, v_9)$	$(u_3; v_3, v_4, v_{10})$	$(u_4; v_5, v_{11}, v_{12})$	$(u_6; v_6, v_{13}, v_{14})$
$\{0,2,3,8\}$	$(u_0; v_0, v_2, v_8)$	$(u_1; v_1, v_3, v_9)$	$(u_2; v_4, v_5, v_{10})$	$(u_4; v_6, v_7, v_{12})$	$(u_{11}; v_{11}, v_{13}, v_{14})$
$\{0,2,4,8\}$	$(u_0; v_0, v_4, v_8)$	$(u_1; v_1, v_3, v_9)$	$(u_3; v_5, v_7, v_{11})$	$(u_{10}; v_{10}, v_{12}, v_{14})$	$(u_{13}; v_2, v_6, v_{13})$
$\{0,2,5,8\}$	$(u_0; v_0, v_5, v_8)$	$(u_1; v_3, v_6, v_9)$	$(u_2; v_4, v_7, v_{10})$	$(u_{11}; v_1, v_{11}, v_{13})$	$(u_{12}; v_2, v_{12}, v_{14})$
$\{0,2,6,8\}$	$(u_0; v_0, v_2, v_6)$	$(u_1; v_1, v_3, v_9)$	$(u_2; v_4, v_8, v_{10})$	$(u_5; v_7, v_{11}, v_{13})$	$(u_{12}; v_5, v_{12}, v_{14})$
$\{0,2,7,8\}$	$(u_0; v_0, v_2, v_7)$	$(u_1; v_1, v_3, v_9)$	$(u_3; v_5, v_{10}, v_{11})$	$(u_4; v_4, v_6, v_{12})$	$(u_6; v_8, v_{13}, v_{14})$
$\{0,3,4,8\}$	$(u_0; v_0, v_3, v_8)$	$(u_1; v_1, v_4, v_5)$	$(u_3; v_6, v_7, v_{11})$	$(u_6; v_9, v_{10}, v_{14})$	$(u_9; v_2, v_{12}, v_{13})$
$\{0,3,5,8\}$	$(u_0; v_0, v_3, v_5)$	$(u_1; v_1, v_4, v_6)$	$(u_2; v_2, v_7, v_{10})$	$(u_8; v_8, v_{11}, v_{13})$	$(u_9; v_9, v_{12}, v_{14})$
$\{0,3,6,8\}$	$(u_0; v_0, v_3, v_6)$	$(u_1; v_1, v_4, v_7)$	$(u_2; v_2, v_5, v_{10})$	$(u_5; v_8, v_{11}, v_{13})$	$(u_6; v_9, v_{12}, v_{14})$
$\{0,3,7,8\}$	$(u_0; v_0, v_3, v_8)$	$(u_1; v_1, v_4, v_9)$	$(u_2; v_2, v_5, v_{10})$	$(u_4; v_7, v_{11}, v_{12})$	$(u_6; v_6, v_{13}, v_{14})$
$\{0,4,5,8\}$	$(u_0; v_0, v_4, v_8)$	$(u_1; v_1, v_5, v_9)$	$(u_6; v_6, v_{10}, v_{14})$	$(u_7; v_7, v_{11}, v_{12})$	$(u_{13}; v_2, v_3, v_{13})$

Table 6.4 – *Continued on next page*

Table 6.4 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5
$\{0,4,6,8\}$	$(u_0; v_4, v_6, v_8)$	$(u_1; v_1, v_5, v_7)$	$(u_3; v_3, v_9, v_{11})$	$(u_6; v_{10}, v_{12}, v_{14})$	$(u_9; v_0, v_2, v_{13})$
$\{0,4,7,8\}$	$(u_0; v_0, v_4, v_7)$	$(u_1; v_1, v_5, v_9)$	$(u_4; v_8, v_{11}, v_{12})$	$(u_6; v_6, v_{10}, v_{13})$	$(u_{10}; v_2, v_3, v_{14})$
$\{0,5,6,8\}$	$(u_0; v_0, v_5, v_6)$	$(u_1; v_1, v_7, v_9)$	$(u_2; v_2, v_8, v_{10})$	$(u_6; v_{11}, v_{12}, v_{14})$	$(u_{13}; v_3, v_4, v_{13})$
$\{0,5,7,8\}$	$(u_0; v_0, v_5, v_7)$	$(u_1; v_1, v_6, v_9)$	$(u_3; v_3, v_8, v_{10})$	$(u_6; v_{11}, v_{13}, v_{14})$	$(u_{12}; v_2, v_4, v_{12})$
$\{0,6,7,8\}$	$(u_0; v_0, v_6, v_7)$	$(u_2; v_2, v_9, v_{10})$	$(u_5; v_5, v_{12}, v_{13})$	$(u_8; v_1, v_8, v_{14})$	$(u_{11}; v_3, v_4, v_{11})$
$\{0,1,3,9\}$	$(u_0; v_0, v_3, v_9)$	$(u_1; v_1, v_2, v_{10})$	$(u_4; v_4, v_7, v_{13})$	$(u_5; v_5, v_6, v_8)$	$(u_{11}; v_{11}, v_{12}, v_{14})$
$\{0,1,4,9\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_5, v_{10})$	$(u_2; v_3, v_6, v_{11})$	$(u_8; v_8, v_9, v_{12})$	$(u_{13}; v_7, v_{13}, v_{14})$
$\{0,1,5,9\}$	$(u_0; v_0, v_5, v_9)$	$(u_1; v_1, v_6, v_{10})$	$(u_2; v_2, v_3, v_{11})$	$(u_3; v_4, v_8, v_{12})$	$(u_{13}; v_7, v_{13}, v_{14})$
$\{0,1,6,9\}$	$(u_0; v_0, v_6, v_9)$	$(u_1; v_1, v_7, v_{10})$	$(u_2; v_3, v_8, v_{11})$	$(u_{11}; v_2, v_5, v_{12})$	$(u_{13}; v_4, v_{13}, v_{14})$
$\{0,1,7,9\}$	$(u_0; v_0, v_1, v_7)$	$(u_1; v_2, v_8, v_{10})$	$(u_2; v_3, v_9, v_{11})$	$(u_4; v_4, v_5, v_{13})$	$(u_5; v_6, v_{12}, v_{14})$
$\{0,2,3,9\}$	$(u_0; v_0, v_2, v_9)$	$(u_1; v_1, v_4, v_{10})$	$(u_3; v_3, v_6, v_{12})$	$(u_5; v_5, v_7, v_8)$	$(u_{11}; v_{11}, v_{13}, v_{14})$
$\{0,2,4,9\}$	$(u_0; v_0, v_2, v_4)$	$(u_1; v_1, v_3, v_5)$	$(u_4; v_6, v_8, v_{13})$	$(u_7; v_7, v_9, v_{11})$	$(u_{10}; v_{10}, v_{12}, v_{14})$
$\{0,2,5,9\}$	$(u_0; v_0, v_5, v_9)$	$(u_1; v_1, v_3, v_{10})$	$(u_2; v_4, v_7, v_{11})$	$(u_8; v_2, v_8, v_{13})$	$(u_{12}; v_6, v_{12}, v_{14})$

Table 6.4 – *Continued on next page*

Table 6.4 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5
$\{0,2,6,9\}$	$(u_0; v_0, v_6, v_9)$	$(u_1; v_1, v_7, v_{10})$	$(u_2; v_4, v_8, v_{11})$	$(u_{11}; v_2, v_5, v_{13})$	$(u_{12}; v_3, v_{12}, v_{14})$
$\{0,2,7,9\}$	$(u_0; v_0, v_2, v_7)$	$(u_1; v_1, v_3, v_{10})$	$(u_2; v_4, v_9, v_{11})$	$(u_5; v_5, v_{12}, v_{14})$	$(u_6; v_6, v_8, v_{13})$
$\{0,2,8,9\}$	$(u_0; v_0, v_8, v_9)$	$(u_1; v_1, v_3, v_{10})$	$(u_2; v_2, v_4, v_{11})$	$(u_4; v_6, v_{12}, v_{13})$	$(u_5; v_5, v_7, v_{14})$
$\{0,3,4,9\}$	$(u_0; v_0, v_3, v_9)$	$(u_1; v_1, v_4, v_5)$	$(u_3; v_6, v_7, v_{12})$	$(u_8; v_2, v_8, v_{11})$	$(u_{10}; v_{10}, v_{13}, v_{14})$
$\{0,3,5,9\}$	$(u_0; v_0, v_3, v_5)$	$(u_1; v_1, v_6, v_{10})$	$(u_4; v_4, v_7, v_{13})$	$(u_8; v_2, v_8, v_{11})$	$(u_9; v_9, v_{12}, v_{14})$
$\{0,3,6,9\}$	Three-component graphs, no decomposition				
$\{0,3,7,9\}$	$(u_0; v_0, v_3, v_7)$	$(u_1; v_1, v_4, v_{10})$	$(u_2; v_2, v_5, v_{11})$	$(u_5; v_8, v_{12}, v_{14})$	$(u_6; v_6, v_9, v_{13})$
$\{0,3,8,9\}$	$(u_0; v_0, v_3, v_8)$	$(u_1; v_1, v_4, v_{10})$	$(u_2; v_2, v_5, v_{11})$	$(u_4; v_7, v_{12}, v_{13})$	$(u_6; v_6, v_9, v_{14})$
$\{0,4,5,9\}$	$(u_0; v_0, v_4, v_5)$	$(u_1; v_1, v_6, v_{10})$	$(u_2; v_2, v_7, v_{11})$	$(u_3; v_3, v_8, v_{12})$	$(u_9; v_9, v_{13}, v_{14})$
$\{0,4,6,9\}$	$(u_0; v_0, v_4, v_6)$	$(u_1; v_1, v_5, v_{10})$	$(u_3; v_3, v_9, v_{12})$	$(u_7; v_7, v_{11}, v_{13})$	$(u_8; v_2, v_8, v_{14})$
$\{0,4,7,9\}$	$(u_0; v_0, v_4, v_7)$	$(u_1; v_1, v_8, v_{10})$	$(u_2; v_2, v_6, v_{11})$	$(u_5; v_5, v_{12}, v_{14})$	$(u_9; v_3, v_9, v_{13})$
$\{0,4,8,9\}$	$(u_0; v_0, v_4, v_8)$	$(u_1; v_1, v_5, v_{10})$	$(u_3; v_3, v_{11}, v_{12})$	$(u_5; v_9, v_{13}, v_{14})$	$(u_{13}; v_2, v_6, v_7)$
$\{0,5,6,9\}$	$(u_0; v_0, v_5, v_6)$	$(u_2; v_2, v_7, v_{11})$	$(u_3; v_3, v_9, v_{12})$	$(u_8; v_8, v_{13}, v_{14})$	$(u_{10}; v_1, v_4, v_{10})$

Table 6.4 – *Continued on next page*

Table 6.4 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5
$\{0,5,7,9\}$	$(u_0; v_0, v_5, v_7)$	$(u_1; v_1, v_8, v_{10})$	$(u_4; v_4, v_{11}, v_{13})$	$(u_9; v_3, v_9, v_{14})$	$(u_{12}; v_2, v_6, v_{12})$
$\{0,5,8,9\}$	$(u_0; v_0, v_5, v_8)$	$(u_1; v_1, v_6, v_{10})$	$(u_2; v_2, v_7, v_{11})$	$(u_4; v_4, v_{12}, v_{13})$	$(u_9; v_3, v_9, v_{14})$
$\{0,6,7,9\}$	$(u_0; v_0, v_6, v_9)$	$(u_1; v_1, v_7, v_8)$	$(u_4; v_4, v_{10}, v_{13})$	$(u_5; v_5, v_{12}, v_{14})$	$(u_{11}; v_2, v_3, v_{11})$
$\{0,6,8,9\}$	$(u_0; v_0, v_6, v_8)$	$(u_1; v_1, v_7, v_{10})$	$(u_3; v_3, v_9, v_{12})$	$(u_5; v_5, v_{13}, v_{14})$	$(u_{11}; v_2, v_4, v_{11})$
$\{0,1,5,10\}$	$(u_0; v_0, v_1, v_{10})$	$(u_1; v_2, v_6, v_{11})$	$(u_2; v_3, v_7, v_{12})$	$(u_3; v_4, v_8, v_{13})$	$(u_4; v_5, v_9, v_{14})$
$\{0,1,6,10\}$	$(u_0; v_0, v_1, v_6)$	$(u_1; v_2, v_7, v_{11})$	$(u_2; v_3, v_8, v_{12})$	$(u_3; v_4, v_9, v_{13})$	$(u_4; v_5, v_{10}, v_{14})$
$\{0,2,5,10\}$	$(u_0; v_0, v_2, v_{10})$	$(u_1; v_1, v_3, v_{11})$	$(u_2; v_4, v_7, v_{12})$	$(u_3; v_5, v_8, v_{13})$	$(u_4; v_6, v_9, v_{14})$
$\{0,2,6,10\}$	$(u_0; v_0, v_2, v_6)$	$(u_1; v_1, v_7, v_{11})$	$(u_2; v_4, v_8, v_{12})$	$(u_3; v_5, v_9, v_{13})$	$(u_8; v_3, v_{10}, v_{14})$
$\{0,2,7,10\}$	$(u_0; v_0, v_2, v_7)$	$(u_1; v_1, v_3, v_8)$	$(u_2; v_4, v_9, v_{12})$	$(u_3; v_5, v_{10}, v_{13})$	$(u_4; v_6, v_{11}, v_{14})$
$\{0,3,5,10\}$	$(u_0; v_0, v_3, v_{10})$	$(u_1; v_1, v_4, v_{11})$	$(u_2; v_2, v_5, v_{12})$	$(u_3; v_6, v_8, v_{13})$	$(u_4; v_7, v_9, v_{14})$
$\{0,3,6,10\}$	$(u_0; v_0, v_6, v_{10})$	$(u_1; v_1, v_4, v_7)$	$(u_2; v_2, v_5, v_{12})$	$(u_3; v_3, v_9, v_{13})$	$(u_8; v_8, v_{11}, v_{14})$
$\{0,3,7,10\}$	$(u_0; v_0, v_3, v_{10})$	$(u_1; v_1, v_4, v_8)$	$(u_2; v_2, v_5, v_{12})$	$(u_4; v_7, v_{11}, v_{14})$	$(u_6; v_6, v_9, v_{13})$
$\{0,3,8,10\}$	$(u_0; v_0, v_3, v_8)$	$(u_1; v_1, v_4, v_9)$	$(u_2; v_2, v_5, v_{10})$	$(u_3; v_6, v_{11}, v_{13})$	$(u_4; v_7, v_{12}, v_{14})$

Table 6.4 – *Continued on next page*

Table 6.4 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5
$\{0,4,5,10\}$	$(u_0; v_0, v_4, v_{10})$	$(u_1; v_1, v_5, v_{11})$	$(u_2; v_2, v_6, v_{12})$	$(u_3; v_3, v_7, v_{13})$	$(u_4; v_8, v_9, v_{14})$
$\{0,4,6,10\}$	$(u_0; v_0, v_4, v_{10})$	$(u_1; v_1, v_5, v_{11})$	$(u_2; v_2, v_6, v_8)$	$(u_3; v_7, v_9, v_{13})$	$(u_8; v_3, v_{12}, v_{14})$
$\{0,4,7,10\}$	$(u_0; v_0, v_4, v_{10})$	$(u_1; v_1, v_5, v_8)$	$(u_2; v_6, v_9, v_{12})$	$(u_3; v_3, v_7, v_{13})$	$(u_7; v_2, v_{11}, v_{14})$
$\{0,4,8,10\}$	$(u_0; v_0, v_4, v_8)$	$(u_1; v_5, v_9, v_{11})$	$(u_2; v_2, v_6, v_{12})$	$(u_3; v_3, v_7, v_{13})$	$(u_6; v_1, v_{10}, v_{14})$
$\{0,4,9,10\}$	$(u_0; v_0, v_4, v_9)$	$(u_1; v_1, v_5, v_{10})$	$(u_2; v_2, v_6, v_{11})$	$(u_3; v_3, v_7, v_{12})$	$(u_4; v_8, v_{13}, v_{14})$
$\{0,5,6,10\}$	$(u_0; v_0, v_5, v_6)$	$(u_1; v_1, v_7, v_{11})$	$(u_2; v_2, v_8, v_{12})$	$(u_3; v_3, v_9, v_{13})$	$(u_4; v_4, v_{10}, v_{14})$
$\{0,5,7,10\}$	$(u_0; v_0, v_5, v_7)$	$(u_1; v_1, v_6, v_8)$	$(u_2; v_2, v_9, v_{12})$	$(u_3; v_3, v_{10}, v_{13})$	$(u_4; v_4, v_{11}, v_{14})$
$\{0,5,8,10\}$	$(u_0; v_0, v_5, v_8)$	$(u_1; v_1, v_6, v_9)$	$(u_2; v_2, v_7, v_{10})$	$(u_3; v_3, v_{11}, v_{13})$	$(u_4; v_4, v_{12}, v_{14})$
$\{0,5,9,10\}$	$(u_0; v_0, v_5, v_9)$	$(u_1; v_1, v_6, v_{10})$	$(u_2; v_2, v_7, v_{11})$	$(u_3; v_3, v_8, v_{12})$	$(u_4; v_4, v_{13}, v_{14})$
$\{0,3,7,11\}$	$(u_0; v_0, v_7, v_{11})$	$(u_1; v_1, v_4, v_8)$	$(u_3; v_3, v_{10}, v_{14})$	$(u_6; v_2, v_6, v_{13})$	$(u_9; v_5, v_9, v_{12})$
$\{0,4,7,11\}$	$(u_0; v_0, v_4, v_{11})$	$(u_1; v_1, v_8, v_{12})$	$(u_3; v_3, v_7, v_{14})$	$(u_6; v_2, v_6, v_{10})$	$(u_9; v_5, v_9, v_{13})$
$\{0,4,8,11\}$	$(u_0; v_0, v_4, v_8)$	$(u_1; v_1, v_5, v_{12})$	$(u_3; v_3, v_7, v_{11})$	$(u_6; v_6, v_{10}, v_{14})$	$(u_9; v_2, v_9, v_{13})$

Table 6.4: S_3-cover of Partite Set V for $n = 15$

6.2.5 S_3-cover of partite set V for $n = 18$

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,1,2,3\}$	$(u_0; v_0, v_1, v_3)$	$(u_2; v_2, v_4, v_5)$	$(u_5; v_6, v_7, v_8)$	$(u_8; v_9, v_{10}, v_{11})$	$(u_{11}; v_{12}, v_{13}, v_{14})$	$(u_{14}; v_{15}, v_{16}, v_{17})$
$\{0,1,2,4\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_3, v_5)$	$(u_6; v_6, v_7, v_{10})$	$(u_7; v_8, v_9, v_{11})$	$(u_{12}; v_{12}, v_{13}, v_{16})$	$(u_{13}; v_{14}, v_{15}, v_{17})$
$\{0,1,3,4\}$	$(u_0; v_0, v_3, v_4)$	$(u_1; v_1, v_2, v_5)$	$(u_5; v_6, v_8, v_9)$	$(u_7; v_7, v_{10}, v_{11})$	$(u_{11}; v_{12}, v_{14}, v_{15})$	$(u_{13}; v_{13}, v_{16}, v_{17})$
$\{0,2,3,4\}$	$(u_0; v_0, v_2, v_3)$	$(u_1; v_1, v_4, v_5)$	$(u_4; v_6, v_7, v_8)$	$(u_7; v_9, v_{10}, v_{11})$	$(u_{10}; v_{12}, v_{13}, v_{14})$	$(u_{13}; v_{15}, v_{16}, v_{17})$
$\{0,1,2,5\}$	$(u_0; v_0, v_1, v_5)$	$(u_2; v_2, v_3, v_4)$	$(u_6; v_6, v_7, v_{11})$	$(u_8; v_8, v_9, v_{10})$	$(u_{12}; v_{12}, v_{13}, v_{17})$	$(u_{14}; v_{14}, v_{15}, v_{16})$
$\{0,1,3,5\}$	$(u_0; v_0, v_1, v_3)$	$(u_1; v_2, v_4, v_6)$	$(u_4; v_5, v_7, v_9)$	$(u_7; v_8, v_{10}, v_{12})$	$(u_{11}; v_{11}, v_{14}, v_{16})$	$(u_{12}; v_{13}, v_{15}, v_{17})$
$\{0,1,4,5\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_5, v_6)$	$(u_3; v_3, v_7, v_8)$	$(u_9; v_9, v_{10}, v_{13})$	$(u_{10}; v_{11}, v_{14}, v_{15})$	$(u_{12}; v_{12}, v_{16}, v_{17})$
$\{0,2,3,5\}$	$(u_0; v_0, v_2, v_5)$	$(u_1; v_1, v_3, v_4)$	$(u_4; v_6, v_7, v_9)$	$(u_8; v_8, v_{10}, v_{11})$	$(u_{10}; v_{12}, v_{13}, v_{15})$	$(u_{14}; v_{14}, v_{16}, v_{17})$
$\{0,2,4,5\}$	$(u_0; v_0, v_2, v_4)$	$(u_1; v_1, v_3, v_6)$	$(u_3; v_5, v_7, v_8)$	$(u_9; v_9, v_{11}, v_{13})$	$(u_{10}; v_{10}, v_{12}, v_{15})$	$(u_{12}; v_{14}, v_{16}, v_{17})$
$\{0,3,4,5\}$	$(u_0; v_0, v_3, v_4)$	$(u_2; v_2, v_6, v_7)$	$(u_5; v_5, v_9, v_{10})$	$(u_8; v_8, v_{12}, v_{13})$	$(u_{11}; v_{11}, v_{15}, v_{16})$	$(u_{14}; v_1, v_{14}, v_{17})$
$\{0,1,2,6\}$	$(u_0; v_0, v_2, v_6)$	$(u_1; v_1, v_3, v_7)$	$(u_3; v_4, v_5, v_9)$	$(u_8; v_8, v_{10}, v_{14})$	$(u_{11}; v_{11}, v_{12}, v_{13})$	$(u_{15}; v_{15}, v_{16}, v_{17})$
$\{0,1,3,6\}$	$(u_0; v_0, v_1, v_6)$	$(u_1; v_2, v_4, v_7)$	$(u_2; v_3, v_5, v_8)$	$(u_9; v_9, v_{10}, v_{15})$	$(u_{10}; v_{11}, v_{13}, v_{16})$	$(u_{11}; v_{12}, v_{14}, v_{17})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,1,4,6\}$	$(u_0; v_0, v_1, v_4)$	$(u_1; v_2, v_5, v_7)$	$(u_2; v_3, v_6, v_8)$	$(u_9; v_9, v_{10}, v_{13})$	$(u_{10}; v_{11}, v_{14}, v_{16})$	$(u_{11}; v_{12}, v_{15}, v_{17})$
$\{0,1,5,6\}$	$(u_0; v_0, v_5, v_6)$	$(u_1; v_1, v_2, v_7)$	$(u_3; v_3, v_4, v_9)$	$(u_7; v_8, v_{12}, v_{13})$	$(u_9; v_{10}, v_{14}, v_{15})$	$(u_{11}; v_{11}, v_{16}, v_{17})$
$\{0,2,3,6\}$	$(u_0; v_0, v_2, v_6)$	$(u_1; v_1, v_3, v_7)$	$(u_2; v_4, v_5, v_8)$	$(u_9; v_9, v_{11}, v_{15})$	$(u_{10}; v_{10}, v_{12}, v_{16})$	$(u_{11}; v_{13}, v_{14}, v_{17})$
$\{0,2,4,6\}$	Two-component graph see $n = 9$ and $D = \{0,1,2,3\}$					
$\{0,2,5,6\}$	$(u_0; v_0, v_2, v_5)$	$(u_1; v_1, v_3, v_6)$	$(u_2; v_4, v_7, v_8)$	$(u_9; v_9, v_{11}, v_{14})$	$(u_{10}; v_{10}, v_{12}, v_{15})$	$(u_{11}; v_{13}, v_{16}, v_{17})$
$\{0,3,4,6\}$	$(u_0; v_0, v_3, v_4)$	$(u_1; v_1, v_5, v_7)$	$(u_2; v_2, v_6, v_8)$	$(u_6; v_9, v_{10}, v_{12})$	$(u_{10}; v_{13}, v_{14}, v_{16})$	$(u_{11}; v_{11}, v_{15}, v_{17})$
$\{0,3,5,6\}$	$(u_0; v_0, v_3, v_5)$	$(u_1; v_1, v_4, v_6)$	$(u_2; v_2, v_7, v_8)$	$(u_6; v_9, v_{11}, v_{12})$	$(u_{10}; v_{10}, v_{13}, v_{15})$	$(u_{11}; v_{14}, v_{16}, v_{17})$
$\{0,4,5,6\}$	$(u_0; v_0, v_4, v_5)$	$(u_1; v_1, v_6, v_7)$	$(u_4; v_8, v_9, v_{10})$	$(u_8; v_{12}, v_{13}, v_{14})$	$(u_{11}; v_{11}, v_{16}, v_{17})$	$(u_{15}; v_2, v_3, v_{15})$
$\{0,1,2,7\}$	$(u_0; v_0, v_2, v_7)$	$(u_1; v_1, v_3, v_8)$	$(u_4; v_5, v_6, v_{11})$	$(u_8; v_9, v_{10}, v_{15})$	$(u_{12}; v_{12}, v_{13}, v_{14})$	$(u_{15}; v_4, v_{16}, v_{17})$
$\{0,1,3,7\}$	$(u_0; v_0, v_1, v_3)$	$(u_1; v_2, v_4, v_8)$	$(u_4; v_5, v_7, v_{11})$	$(u_6; v_6, v_9, v_{13})$	$(u_9; v_{10}, v_{12}, v_{16})$	$(u_{14}; v_{14}, v_{15}, v_{17})$
$\{0,1,4,7\}$	$(u_0; v_0, v_4, v_7)$	$(u_1; v_1, v_2, v_5)$	$(u_2; v_3, v_6, v_9)$	$(u_8; v_8, v_{12}, v_{15})$	$(u_9; v_{10}, v_{13}, v_{16})$	$(u_{10}; v_{11}, v_{14}, v_{17})$
$\{0,1,5,7\}$	$(u_0; v_0, v_5, v_7)$	$(u_1; v_1, v_2, v_6)$	$(u_3; v_3, v_4, v_8)$	$(u_9; v_9, v_{14}, v_{16})$	$(u_{10}; v_{10}, v_{11}, v_{15})$	$(u_{12}; v_{12}, v_{13}, v_{17})$
$\{0,1,6,7\}$	$(u_0; v_0, v_1, v_6)$	$(u_2; v_2, v_3, v_8)$	$(u_4; v_4, v_5, v_{10})$	$(u_6; v_7, v_{12}, v_{13})$	$(u_8; v_9, v_{14}, v_{15})$	$(u_{10}; v_{11}, v_{16}, v_{17})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,2,3,7\}$	$(u_0;v_0,v_2,v_7)$	$(u_1;v_1,v_4,v_8)$	$(u_3;v_3,v_5,v_6)$	$(u_9;v_9,v_{11},v_{16})$	$(u_{10};v_{10},v_{13},v_{17})$	$(u_{12};v_{12},v_{14},v_{15})$
$\{0,2,4,7\}$	$(u_0;v_0,v_2,v_7)$	$(u_1;v_1,v_3,v_5)$	$(u_2;v_4,v_6,v_9)$	$(u_8;v_8,v_{10},v_{15})$	$(u_9;v_{11},v_{13},v_{16})$	$(u_{10};v_{12},v_{14},v_{17})$
$\{0,2,5,7\}$	$(u_0;v_0,v_2,v_5)$	$(u_1;v_1,v_3,v_6)$	$(u_2;v_4,v_7,v_9)$	$(u_8;v_8,v_{10},v_{13})$	$(u_9;v_{11},v_{14},v_{16})$	$(u_{10};v_{12},v_{15},v_{17})$
$\{0,2,6,7\}$	$(u_0;v_0,v_2,v_6)$	$(u_1;v_1,v_3,v_8)$	$(u_5;v_5,v_7,v_{11})$	$(u_7;v_9,v_{13},v_{14})$	$(u_{10};v_{10},v_{12},v_{16})$	$(u_{15};v_4,v_{15},v_{17})$
$\{0,3,4,7\}$	$(u_0;v_0,v_3,v_4)$	$(u_1;v_1,v_5,v_8)$	$(u_2;v_2,v_6,v_9)$	$(u_7;v_7,v_{10},v_{14})$	$(u_8;v_{11},v_{12},v_{15})$	$(u_{13};v_{13},v_{16},v_{17})$
$\{0,3,5,7\}$	$(u_0;v_0,v_3,v_7)$	$(u_1;v_1,v_4,v_6)$	$(u_2;v_2,v_5,v_9)$	$(u_5;v_8,v_{10},v_{12})$	$(u_{10};v_{13},v_{15},v_{17})$	$(u_{11};v_{11},v_{14},v_{16})$
$\{0,3,6,7\}$	$(u_0;v_0,v_3,v_6)$	$(u_1;v_1,v_4,v_7)$	$(u_2;v_2,v_5,v_8)$	$(u_6;v_9,v_{12},v_{13})$	$(u_8;v_{11},v_{14},v_{15})$	$(u_{10};v_{10},v_{16},v_{17})$
$\{0,4,5,7\}$	$(u_0;v_0,v_4,v_7)$	$(u_1;v_1,v_5,v_6)$	$(u_4;v_8,v_9,v_{11})$	$(u_8;v_{12},v_{13},v_{15})$	$(u_{10};v_{10},v_{14},v_{17})$	$(u_{16};v_2,v_3,v_{16})$
$\{0,4,6,7\}$	$(u_0;v_0,v_4,v_6)$	$(u_1;v_1,v_7,v_8)$	$(u_5;v_5,v_{11},v_{12})$	$(u_9;v_9,v_{13},v_{15})$	$(u_{10};v_{10},v_{16},v_{17})$	$(u_{14};v_2,v_3,v_{14})$
$\{0,5,6,7\}$	$(u_0;v_0,v_5,v_6)$	$(u_1;v_1,v_7,v_8)$	$(u_3;v_3,v_9,v_{10})$	$(u_7;v_{12},v_{13},v_{14})$	$(u_{11};v_{11},v_{16},v_{17})$	$(u_{15};v_2,v_4,v_{15})$
$\{0,1,2,8\}$	$(u_0;v_0,v_1,v_8)$	$(u_1;v_2,v_3,v_9)$	$(u_4;v_4,v_5,v_{12})$	$(u_5;v_6,v_7,v_{13})$	$(u_9;v_{10},v_{11},v_{17})$	$(u_{14};v_{14},v_{15},v_{16})$
$\{0,1,3,8\}$	$(u_0;v_0,v_3,v_8)$	$(u_1;v_1,v_2,v_9)$	$(u_4;v_5,v_7,v_{12})$	$(u_{10};v_{10},v_{11},v_{13})$	$(u_{14};v_4,v_{14},v_{15})$	$(u_{16};v_6,v_{16},v_{17})$
$\{0,1,4,8\}$	$(u_0;v_0,v_4,v_8)$	$(u_1;v_1,v_2,v_5)$	$(u_2;v_3,v_6,v_{10})$	$(u_7;v_7,v_{11},v_{15})$	$(u_8;v_9,v_{12},v_{16})$	$(u_{13};v_{13},v_{14},v_{17})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,1,5,8\}$	$(u_0;v_0,v_5,v_8)$	$(u_1;v_1,v_6,v_9)$	$(u_2;v_3,v_7,v_{10})$	$(u_{11};v_{11},v_{12},v_{16})$	$(u_{12};v_2,v_{13},v_{17})$	$(u_{14};v_4,v_{14},v_{15})$
$\{0,1,6,8\}$	$(u_0;v_0,v_6,v_8)$	$(u_1;v_1,v_2,v_7)$	$(u_4;v_4,v_5,v_{12})$	$(u_9;v_9,v_{15},v_{17})$	$(u_{10};v_{10},v_{11},v_{16})$	$(u_{13};v_3,v_{13},v_{14})$
$\{0,1,7,8\}$	$(u_0;v_0,v_1,v_7)$	$(u_1;v_2,v_8,v_9)$	$(u_3;v_3,v_{10},v_{11})$	$(u_5;v_5,v_{12},v_{13})$	$(u_{14};v_4,v_{14},v_{15})$	$(u_{16};v_6,v_{16},v_{17})$
$\{0,2,3,8\}$	$(u_0;v_0,v_2,v_8)$	$(u_1;v_1,v_3,v_4)$	$(u_3;v_5,v_6,v_{11})$	$(u_7;v_7,v_9,v_{15})$	$(u_{10};v_{10},v_{12},v_{13})$	$(u_{14};v_{14},v_{16},v_{17})$
$\{0,2,4,8\}$	Two-component graph see $n=9$ and $D=\{0,1,2,4\}$					
$\{0,2,5,8\}$	$(u_0;v_0,v_5,v_8)$	$(u_1;v_1,v_3,v_6)$	$(u_2;v_2,v_4,v_7)$	$(u_7;v_9,v_{12},v_{15})$	$(u_8;v_{10},v_{13},v_{16})$	$(u_9;v_{11},v_{14},v_{17})$
$\{0,2,6,8\}$	Two-component graph see $n=9$ and $D=\{0,1,3,4\}$					
$\{0,2,7,8\}$	$(u_0;v_0,v_2,v_7)$	$(u_1;v_1,v_3,v_9)$	$(u_4;v_4,v_{11},v_{12})$	$(u_6;v_6,v_{13},v_{14})$	$(u_8;v_8,v_{10},v_{16})$	$(u_{15};v_5,v_{15},v_{17})$
$\{0,3,4,8\}$	$(u_0;v_0,v_3,v_8)$	$(u_1;v_1,v_5,v_9)$	$(u_2;v_2,v_6,v_{10})$	$(u_4;v_4,v_7,v_{12})$	$(u_{11};v_{11},v_{14},v_{15})$	$(u_{13};v_{13},v_{16},v_{17})$
$\{0,3,5,8\}$	$(u_0;v_0,v_3,v_5)$	$(u_1;v_1,v_4,v_6)$	$(u_2;v_2,v_7,v_{10})$	$(u_6;v_9,v_{11},v_{14})$	$(u_8;v_8,v_{13},v_{16})$	$(u_{12};v_{12},v_{15},v_{17})$
$\{0,3,6,8\}$	$(u_0;v_0,v_3,v_6)$	$(u_1;v_1,v_4,v_7)$	$(u_2;v_2,v_5,v_8)$	$(u_7;v_{10},v_{13},v_{15})$	$(u_8;v_{11},v_{14},v_{16})$	$(u_9;v_9,v_{12},v_{17})$
$\{0,3,7,8\}$	$(u_0;v_0,v_3,v_7)$	$(u_1;v_1,v_4,v_8)$	$(u_2;v_2,v_5,v_{10})$	$(u_6;v_6,v_{13},v_{14})$	$(u_8;v_{11},v_{15},v_{16})$	$(u_9;v_9,v_{12},v_{17})$
$\{0,4,5,8\}$	$(u_0;v_0,v_4,v_8)$	$(u_1;v_1,v_5,v_9)$	$(u_2;v_2,v_7,v_{10})$	$(u_7;v_{11},v_{12},v_{15})$	$(u_9;v_{13},v_{14},v_{17})$	$(u_{16};v_3,v_6,v_{16})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,4,6,8\}$	Two-component graph see $n=9$ and $D=\{0,2,3,4\}$					
$\{0,4,7,8\}$	$(u_0;v_0,v_4,v_7)$	$(u_1;v_1,v_5,v_9)$	$(u_2;v_2,v_6,v_{10})$	$(u_7;v_{11},v_{14},v_{15})$	$(u_8;v_8,v_{12},v_{16})$	$(u_{13};v_3,v_{13},v_{17})$
$\{0,5,6,8\}$	$(u_0;v_0,v_5,v_6)$	$(u_1;v_1,v_7,v_9)$	$(u_4;v_4,v_{10},v_{12})$	$(u_8;v_8,v_{13},v_{14})$	$(u_{11};v_{11},v_{16},v_{17})$	$(u_{15};v_2,v_3,v_{15})$
$\{0,5,7,8\}$	$(u_0;v_0,v_5,v_7)$	$(u_1;v_1,v_6,v_9)$	$(u_3;v_3,v_8,v_{11})$	$(u_5;v_{10},v_{12},v_{13})$	$(u_9;v_{14},v_{16},v_{17})$	$(u_{15};v_2,v_4,v_{15})$
$\{0,6,7,8\}$	$(u_0;v_0,v_6,v_8)$	$(u_1;v_1,v_7,v_9)$	$(u_3;v_3,v_{10},v_{11})$	$(u_5;v_5,v_{12},v_{13})$	$(u_9;v_{15},v_{16},v_{17})$	$(u_{14};v_2,v_4,v_{14})$
$\{0,1,2,9\}$	$(u_0;v_0,v_1,v_9)$	$(u_2;v_3,v_4,v_{11})$	$(u_5;v_5,v_6,v_7)$	$(u_8;v_8,v_{10},v_{17})$	$(u_{11};v_2,v_{12},v_{13})$	$(u_{14};v_{14},v_{15},v_{16})$
$\{0,1,3,9\}$	$(u_0;v_0,v_1,v_9)$	$(u_1;v_2,v_4,v_{10})$	$(u_2;v_3,v_5,v_{11})$	$(u_5;v_6,v_8,v_{14})$	$(u_{12};v_{12},v_{13},v_{15})$	$(u_{16};v_7,v_{16},v_{17})$
$\{0,1,4,9\}$	$(u_0;v_0,v_4,v_9)$	$(u_1;v_1,v_5,v_{10})$	$(u_2;v_2,v_3,v_6)$	$(u_7;v_7,v_8,v_{16})$	$(u_{11};v_{11},v_{12},v_{15})$	$(u_{13};v_{13},v_{14},v_{17})$
$\{0,1,5,9\}$	$(u_0;v_0,v_5,v_9)$	$(u_1;v_1,v_2,v_{10})$	$(u_3;v_3,v_8,v_{12})$	$(u_6;v_6,v_{11},v_{15})$	$(u_{13};v_4,v_{13},v_{14})$	$(u_{16};v_7,v_{16},v_{17})$
$\{0,1,6,9\}$	$(u_0;v_0,v_6,v_9)$	$(u_1;v_1,v_7,v_{10})$	$(u_2;v_2,v_8,v_{11})$	$(u_{12};v_3,v_{12},v_{13})$	$(u_{14};v_5,v_{14},v_{15})$	$(u_{16};v_4,v_{16},v_{17})$
$\{0,1,7,9\}$	$(u_0;v_0,v_1,v_9)$	$(u_1;v_2,v_8,v_{10})$	$(u_4;v_4,v_{11},v_{13})$	$(u_5;v_6,v_{12},v_{14})$	$(u_{14};v_3,v_5,v_{15})$	$(u_{16};v_7,v_{16},v_{17})$
$\{0,1,8,9\}$	$(u_0;v_0,v_1,v_8)$	$(u_1;v_2,v_9,v_{10})$	$(u_3;v_3,v_4,v_{11})$	$(u_4;v_5,v_{12},v_{13})$	$(u_6;v_6,v_{14},v_{15})$	$(u_{16};v_7,v_{16},v_{17})$
$\{0,2,3,9\}$	$(u_0;v_0,v_2,v_9)$	$(u_1;v_1,v_3,v_4)$	$(u_3;v_5,v_6,v_{12})$	$(u_5;v_7,v_8,v_{14})$	$(u_8;v_{10},v_{11},v_{17})$	$(u_{13};v_{13},v_{15},v_{16})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,2,4,9\}$	$(u_0; v_0, v_2, v_9)$	$(u_1; v_1, v_3, v_5)$	$(u_4; v_6, v_8, v_{13})$	$(u_7; v_7, v_{11}, v_{16})$	$(u_{10}; v_{10}, v_{12}, v_{14})$	$(u_{13}; v_4, v_{15}, v_{17})$
$\{0,2,5,9\}$	$(u_0; v_0, v_5, v_9)$	$(u_1; v_1, v_3, v_{10})$	$(u_2; v_2, v_4, v_7)$	$(u_6; v_6, v_8, v_{15})$	$(u_{11}; v_{11}, v_{13}, v_{16})$	$(u_{12}; v_{12}, v_{14}, v_{17})$
$\{0,2,6,9\}$	$(u_0; v_0, v_2, v_9)$	$(u_1; v_1, v_3, v_7)$	$(u_4; v_4, v_6, v_{10})$	$(u_6; v_8, v_{12}, v_{15})$	$(u_{11}; v_{11}, v_{13}, v_{17})$	$(u_{14}; v_5, v_{14}, v_{16})$
$\{0,2,7,9\}$	$(u_0; v_0, v_2, v_7)$	$(u_1; v_1, v_3, v_8)$	$(u_2; v_4, v_9, v_{11})$	$(u_6; v_6, v_{13}, v_{15})$	$(u_{10}; v_{10}, v_{12}, v_{17})$	$(u_{14}; v_5, v_{14}, v_{16})$
$\{0,2,8,9\}$	$(u_0; v_0, v_2, v_8)$	$(u_1; v_1, v_9, v_{10})$	$(u_3; v_3, v_{11}, v_{12})$	$(u_5; v_5, v_7, v_{13})$	$(u_{14}; v_4, v_{14}, v_{16})$	$(u_{15}; v_6, v_{15}, v_{17})$
$\{0,3,4,9\}$	$(u_0; v_0, v_3, v_9)$	$(u_1; v_1, v_4, v_{10})$	$(u_2; v_2, v_6, v_{11})$	$(u_4; v_7, v_8, v_{13})$	$(u_{12}; v_{12}, v_{15}, v_{16})$	$(u_{14}; v_5, v_{14}, v_{17})$
$\{0,3,5,9\}$	$(u_0; v_0, v_3, v_9)$	$(u_1; v_1, v_6, v_{10})$	$(u_2; v_2, v_7, v_{11})$	$(u_5; v_5, v_8, v_{14})$	$(u_{12}; v_{12}, v_{15}, v_{17})$	$(u_{13}; v_4, v_{13}, v_{16})$
$\{0,3,6,9\}$	Three-component graph see $n = 6$ and $D = \{0,1,2,3\}$					
$\{0,3,7,9\}$	$(u_0; v_0, v_7, v_9)$	$(u_1; v_1, v_4, v_8)$	$(u_3; v_3, v_6, v_{10})$	$(u_5; v_5, v_{12}, v_{14})$	$(u_8; v_{11}, v_{15}, v_{17})$	$(u_{13}; v_2, v_{13}, v_{16})$
$\{0,3,8,9\}$	$(u_0; v_0, v_3, v_8)$	$(u_1; v_1, v_4, v_9)$	$(u_2; v_2, v_5, v_{10})$	$(u_4; v_7, v_{12}, v_{13})$	$(u_6; v_6, v_{14}, v_{15})$	$(u_8; v_{11}, v_{16}, v_{17})$
$\{0,4,5,9\}$	$(u_0; v_0, v_4, v_9)$	$(u_1; v_1, v_5, v_6)$	$(u_2; v_2, v_7, v_{11})$	$(u_8; v_8, v_{12}, v_{13})$	$(u_{10}; v_{10}, v_{14}, v_{15})$	$(u_{12}; v_3, v_{16}, v_{17})$
$\{0,4,6,9\}$	$(u_0; v_0, v_4, v_9)$	$(u_1; v_1, v_5, v_{10})$	$(u_2; v_2, v_8, v_{11})$	$(u_7; v_7, v_{13}, v_{16})$	$(u_8; v_{12}, v_{14}, v_{17})$	$(u_{15}; v_3, v_6, v_{15})$
$\{0,4,7,9\}$	$(u_0; v_0, v_4, v_7)$	$(u_1; v_1, v_5, v_8)$	$(u_2; v_2, v_9, v_{11})$	$(u_6; v_6, v_{13}, v_{15})$	$(u_{10}; v_{10}, v_{14}, v_{17})$	$(u_{12}; v_3, v_{12}, v_{16})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,4,8,9\}$	$(u_0;v_0,v_4,v_8)$	$(u_1;v_1,v_5,v_{10})$	$(u_3;v_3,v_7,v_{11})$	$(u_6;v_6,v_{14},v_{15})$	$(u_9;v_9,v_{13},v_{17})$	$(u_{12};v_2,v_{12},v_{16})$
$\{0,5,6,9\}$	$(u_0;v_0,v_5,v_9)$	$(u_1;v_1,v_6,v_{10})$	$(u_2;v_2,v_7,v_8)$	$(u_6;v_{11},v_{12},v_{15})$	$(u_8;v_{13},v_{14},v_{17})$	$(u_{16};v_3,v_4,v_{16})$
$\{0,5,7,9\}$	$(u_0;v_0,v_5,v_9)$	$(u_1;v_1,v_8,v_{10})$	$(u_4;v_4,v_{11},v_{13})$	$(u_7;v_7,v_{14},v_{16})$	$(u_{12};v_3,v_{12},v_{17})$	$(u_{15};v_2,v_6,v_{15})$
$\{0,5,8,9\}$	$(u_0;v_0,v_5,v_8)$	$(u_1;v_1,v_6,v_{10})$	$(u_2;v_2,v_7,v_{11})$	$(u_7;v_{12},v_{15},v_{16})$	$(u_9;v_9,v_{14},v_{17})$	$(u_{13};v_3,v_4,v_{13})$
$\{0,6,7,9\}$	$(u_0;v_0,v_6,v_9)$	$(u_1;v_1,v_8,v_{10})$	$(u_5;v_5,v_{12},v_{14})$	$(u_7;v_7,v_{13},v_{16})$	$(u_{11};v_2,v_{11},v_{17})$	$(u_{15};v_3,v_4,v_{15})$
$\{0,6,8,9\}$	$(u_0;v_0,v_6,v_8)$	$(u_1;v_1,v_7,v_9)$	$(u_2;v_2,v_{10},v_{11})$	$(u_4;v_4,v_{12},v_{13})$	$(u_8;v_{14},v_{16},v_{17})$	$(u_{15};v_3,v_5,v_{15})$
$\{0,7,8,9\}$	$(u_0;v_0,v_7,v_9)$	$(u_2;v_2,v_{10},v_{11})$	$(u_5;v_5,v_{13},v_{14})$	$(u_8;v_8,v_{16},v_{17})$	$(u_{12};v_1,v_3,v_{12})$	$(u_{15};v_4,v_6,v_{15})$
$\{0,1,2,10\}$	$(u_0;v_0,v_1,v_{10})$	$(u_1;v_2,v_3,v_{11})$	$(u_4;v_4,v_5,v_6)$	$(u_7;v_7,v_8,v_9)$	$(u_{12};v_{12},v_{13},v_{14})$	$(u_{15};v_{15},v_{16},v_{17})$
$\{0,1,3,10\}$	$(u_0;v_0,v_1,v_3)$	$(u_1;v_2,v_4,v_{11})$	$(u_4;v_5,v_7,v_{14})$	$(u_6;v_6,v_9,v_{16})$	$(u_7;v_8,v_{10},v_{17})$	$(u_{12};v_{12},v_{13},v_{15})$
$\{0,1,4,10\}$	$(u_0;v_0,v_4,v_{10})$	$(u_1;v_1,v_5,v_{11})$	$(u_2;v_2,v_3,v_6)$	$(u_8;v_8,v_9,v_{12})$	$(u_{13};v_{13},v_{14},v_{17})$	$(u_{15};v_7,v_{15},v_{16})$
$\{0,1,5,10\}$	$(u_0;v_0,v_1,v_{10})$	$(u_1;v_2,v_6,v_{11})$	$(u_3;v_3,v_4,v_{13})$	$(u_4;v_5,v_9,v_{14})$	$(u_7;v_8,v_{12},v_{17})$	$(u_{15};v_7,v_{15},v_{16})$
$\{0,1,6,10\}$	$(u_0;v_0,v_1,v_6)$	$(u_1;v_2,v_7,v_{11})$	$(u_2;v_3,v_8,v_{12})$	$(u_9;v_9,v_{10},v_{15})$	$(u_{13};v_5,v_{13},v_{14})$	$(u_{16};v_4,v_{16},v_{17})$
$\{0,1,7,10\}$	$(u_0;v_0,v_1,v_7)$	$(u_1;v_2,v_8,v_{11})$	$(u_2;v_3,v_9,v_{12})$	$(u_3;v_4,v_{10},v_{13})$	$(u_{14};v_6,v_{14},v_{15})$	$(u_{16};v_5,v_{16},v_{17})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,1,8,10\}$	$(u_0; v_0, v_1, v_{10})$	$(u_1; v_2, v_9, v_{11})$	$(u_3; v_3, v_4, v_{13})$	$(u_4; v_5, v_{12}, v_{14})$	$(u_6; v_6, v_7, v_{16})$	$(u_7; v_8, v_{15}, v_{17})$
$\{0,1,9,10\}$	$(u_0; v_0, v_1, v_9)$	$(u_1; v_2, v_{10}, v_{11})$	$(u_3; v_3, v_4, v_{12})$	$(u_4; v_5, v_{13}, v_{14})$	$(u_6; v_6, v_7, v_{15})$	$(u_7; v_8, v_{16}, v_{17})$
$\{0,2,3,10\}$	$(u_0; v_0, v_2, v_{10})$	$(u_1; v_1, v_3, v_4)$	$(u_3; v_5, v_6, v_{13})$	$(u_5; v_7, v_8, v_{15})$	$(u_9; v_9, v_{11}, v_{12})$	$(u_{14}; v_{14}, v_{16}, v_{17})$
$\{0,2,4,10\}$	Two-component graph see $n = 9$ and $D = \{0,1,2,5\}$					
$\{0,2,5,10\}$	$(u_0; v_0, v_2, v_{10})$	$(u_1; v_1, v_3, v_{11})$	$(u_4; v_4, v_9, v_{14})$	$(u_6; v_6, v_8, v_{16})$	$(u_7; v_7, v_{12}, v_{17})$	$(u_{13}; v_5, v_{13}, v_{15})$
$\{0,2,6,10\}$	Two-component graph see $n = 9$ and $D = \{0,1,3,5\}$					
$\{0,2,7,10\}$	$(u_0; v_0, v_2, v_{10})$	$(u_1; v_1, v_8, v_{11})$	$(u_2; v_4, v_9, v_{12})$	$(u_3; v_3, v_5, v_{13})$	$(u_{14}; v_6, v_{14}, v_{16})$	$(u_{15}; v_7, v_{15}, v_{17})$
$\{0,2,8,10\}$	Two-component graph see $n = 9$ and $D = \{0,1,4,5\}$					
$\{0,2,9,10\}$	$(u_0; v_0, v_2, v_{10})$	$(u_1; v_1, v_3, v_{11})$	$(u_3; v_5, v_{12}, v_{13})$	$(u_4; v_4, v_6, v_{14})$	$(u_6; v_8, v_{15}, v_{16})$	$(u_7; v_7, v_9, v_{17})$
$\{0,3,4,10\}$	$(u_0; v_0, v_3, v_{10})$	$(u_1; v_1, v_4, v_{11})$	$(u_2; v_2, v_6, v_{12})$	$(u_4; v_7, v_8, v_{14})$	$(u_5; v_5, v_9, v_{15})$	$(u_{13}; v_{13}, v_{16}, v_{17})$
$\{0,3,5,10\}$	$(u_0; v_0, v_5, v_{10})$	$(u_1; v_1, v_6, v_{11})$	$(u_4; v_4, v_9, v_{14})$	$(u_7; v_7, v_{12}, v_{17})$	$(u_{10}; v_2, v_{13}, v_{15})$	$(u_{16}; v_3, v_8, v_{16})$
$\{0,3,6,10\}$	$(u_0; v_0, v_3, v_6)$	$(u_1; v_1, v_4, v_7)$	$(u_2; v_2, v_5, v_8)$	$(u_9; v_9, v_{12}, v_{15})$	$(u_{10}; v_{10}, v_{13}, v_{16})$	$(u_{11}; v_{11}, v_{14}, v_{17})$
$\{0,3,7,10\}$	$(u_0; v_0, v_3, v_7)$	$(u_1; v_1, v_4, v_{11})$	$(u_2; v_2, v_5, v_9)$	$(u_5; v_8, v_{12}, v_{15})$	$(u_6; v_6, v_{13}, v_{16})$	$(u_7; v_{10}, v_{14}, v_{17})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,3,8,10\}$	$(u_0; v_0, v_8, v_{10})$	$(u_1; v_1, v_4, v_{11})$	$(u_2; v_2, v_5, v_{12})$	$(u_3; v_3, v_6, v_{13})$	$(u_6; v_9, v_{14}, v_{16})$	$(u_7; v_7, v_{15}, v_{17})$
$\{0,3,9,10\}$	$(u_0; v_0, v_3, v_9)$	$(u_1; v_1, v_4, v_{10})$	$(u_2; v_2, v_5, v_{11})$	$(u_3; v_6, v_{12}, v_{13})$	$(u_5; v_8, v_{14}, v_{15})$	$(u_7; v_7, v_{16}, v_{17})$
$\{0,4,5,10\}$	$(u_0; v_0, v_4, v_{10})$	$(u_1; v_1, v_6, v_{11})$	$(u_2; v_2, v_7, v_{12})$	$(u_4; v_8, v_9, v_{14})$	$(u_{11}; v_3, v_{15}, v_{16})$	$(u_{13}; v_5, v_{13}, v_{17})$
$\{0,4,6,10\}$	Two-component graph see $n = 9$ and $D = \{0,2,3,5\}$					
$\{0,4,7,10\}$	$(u_0; v_0, v_4, v_{10})$	$(u_1; v_1, v_5, v_8)$	$(u_2; v_2, v_9, v_{12})$	$(u_6; v_6, v_{13}, v_{16})$	$(u_7; v_7, v_{14}, v_{17})$	$(u_{11}; v_3, v_{11}, v_{15})$
$\{0,4,8,10\}$	Two-component graph see $n = 9$ and $D = \{0,2,4,5\}$					
$\{0,4,9,10\}$	$(u_0; v_0, v_4, v_9)$	$(u_1; v_1, v_5, v_{10})$	$(u_2; v_2, v_6, v_{12})$	$(u_4; v_8, v_{13}, v_{14})$	$(u_7; v_7, v_{16}, v_{17})$	$(u_{11}; v_3, v_{11}, v_{15})$
$\{0,5,6,10\}$	$(u_0; v_0, v_5, v_{10})$	$(u_1; v_1, v_6, v_{11})$	$(u_2; v_2, v_7, v_8)$	$(u_7; v_{12}, v_{13}, v_{17})$	$(u_9; v_9, v_{14}, v_{15})$	$(u_{16}; v_3, v_4, v_{16})$
$\{0,5,7,10\}$	$(u_0; v_0, v_5, v_{10})$	$(u_1; v_1, v_6, v_{11})$	$(u_3; v_3, v_8, v_{13})$	$(u_7; v_7, v_{12}, v_{17})$	$(u_9; v_9, v_{14}, v_{16})$	$(u_{15}; v_2, v_4, v_{15})$
$\{0,5,8,10\}$	$(u_0; v_0, v_5, v_{10})$	$(u_1; v_1, v_6, v_9)$	$(u_3; v_3, v_8, v_{13})$	$(u_6; v_{11}, v_{14}, v_{16})$	$(u_7; v_7, v_{12}, v_{15})$	$(u_{12}; v_2, v_4, v_{17})$
$\{0,5,9,10\}$	$(u_0; v_0, v_5, v_9)$	$(u_1; v_1, v_6, v_{11})$	$(u_3; v_3, v_8, v_{12})$	$(u_4; v_4, v_{13}, v_{14})$	$(u_7; v_7, v_{16}, v_{17})$	$(u_{10}; v_2, v_{10}, v_{15})$
$\{0,6,7,10\}$	$(u_0; v_0, v_6, v_{10})$	$(u_1; v_1, v_7, v_{11})$	$(u_2; v_2, v_9, v_{12})$	$(u_7; v_{13}, v_{14}, v_{17})$	$(u_{15}; v_3, v_4, v_{15})$	$(u_{16}; v_5, v_8, v_{16})$
$\{0,6,8,10\}$	Two-component graph see $n = 9$ and $D = \{0,3,4,5\}$					

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,6,9,10\}$	$(u_0;v_0,v_6,v_9)$	$(u_1;v_1,v_7,v_{10})$	$(u_2;v_2,v_8,v_{11})$	$(u_5;v_5,v_{14},v_{15})$	$(u_7;v_{13},v_{16},v_{17})$	$(u_{12};v_3,v_4,v_{12})$
$\{0,7,8,10\}$	$(u_0;v_0,v_7,v_8)$	$(u_1;v_1,v_9,v_{11})$	$(u_3;v_3,v_{10},v_{13})$	$(u_7;v_{14},v_{15},v_{17})$	$(u_{12};v_2,v_4,v_{12})$	$(u_{16};v_5,v_6,v_{16})$
$\{0,7,9,10\}$	$(u_0;v_0,v_7,v_9)$	$(u_1;v_1,v_8,v_{11})$	$(u_3;v_3,v_{10},v_{12})$	$(u_6;v_6,v_{13},v_{15})$	$(u_7;v_{14},v_{16},v_{17})$	$(u_{13};v_2,v_4,v_5)$
$\{0,8,9,10\}$	$(u_0;v_0,v_8,v_9)$	$(u_1;v_1,v_{10},v_{11})$	$(u_4;v_4,v_{13},v_{14})$	$(u_7;v_7,v_{16},v_{17})$	$(u_{12};v_2,v_3,v_{12})$	$(u_{15};v_5,v_6,v_{15})$
$\{0,1,4,11\}$	$(u_0;v_0,v_1,v_{11})$	$(u_1;v_2,v_5,v_{12})$	$(u_3;v_3,v_4,v_{14})$	$(u_6;v_6,v_7,v_{17})$	$(u_9;v_9,v_{10},v_{13})$	$(u_{15};v_8,v_{15},v_{16})$
$\{0,1,5,11\}$	$(u_0;v_0,v_1,v_5)$	$(u_1;v_2,v_6,v_{12})$	$(u_2;v_3,v_7,v_{13})$	$(u_3;v_4,v_8,v_{14})$	$(u_{10};v_{10},v_{11},v_{15})$	$(u_{16};v_9,v_{16},v_{17})$
$\{0,1,6,11\}$	$(u_0;v_0,v_6,v_{11})$	$(u_1;v_1,v_7,v_{12})$	$(u_2;v_2,v_8,v_{13})$	$(u_3;v_3,v_4,v_{14})$	$(u_4;v_5,v_{10},v_{15})$	$(u_{16};v_9,v_{16},v_{17})$
$\{0,1,7,11\}$	$(u_0;v_0,v_1,v_7)$	$(u_1;v_2,v_8,v_{12})$	$(u_3;v_3,v_4,v_{14})$	$(u_4;v_5,v_{11},v_{15})$	$(u_6;v_6,v_{13},v_{17})$	$(u_9;v_9,v_{10},v_{16})$
$\{0,1,8,11\}$	$(u_0;v_0,v_1,v_8)$	$(u_1;v_2,v_9,v_{12})$	$(u_2;v_3,v_{10},v_{13})$	$(u_3;v_4,v_{11},v_{14})$	$(u_6;v_6,v_7,v_{17})$	$(u_{15};v_5,v_{15},v_{16})$
$\{0,2,4,11\}$	$(u_0;v_0,v_2,v_{11})$	$(u_1;v_1,v_3,v_5)$	$(u_2;v_4,v_6,v_{13})$	$(u_5;v_7,v_9,v_{16})$	$(u_{10};v_{10},v_{12},v_{14})$	$(u_{15};v_8,v_{15},v_{17})$
$\{0,2,5,11\}$	$(u_0;v_0,v_2,v_{11})$	$(u_1;v_1,v_3,v_6)$	$(u_4;v_4,v_9,v_{15})$	$(u_5;v_5,v_7,v_{16})$	$(u_8;v_8,v_{10},v_{13})$	$(u_{12};v_{12},v_{14},v_{17})$
$\{0,2,6,11\}$	$(u_0;v_0,v_6,v_{11})$	$(u_1;v_1,v_7,v_{12})$	$(u_2;v_2,v_4,v_{13})$	$(u_3;v_5,v_9,v_{14})$	$(u_{10};v_3,v_{10},v_{16})$	$(u_{15};v_8,v_{15},v_{17})$
$\{0,2,7,11\}$	$(u_0;v_0,v_2,v_7)$	$(u_1;v_1,v_3,v_{12})$	$(u_3;v_5,v_{10},v_{14})$	$(u_4;v_4,v_6,v_{15})$	$(u_6;v_8,v_{13},v_{17})$	$(u_9;v_9,v_{11},v_{16})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,2,8,11\}$	$(u_0;v_0,v_2,v_{11})$	$(u_1;v_1,v_3,v_9)$	$(u_2;v_4,v_{10},v_{13})$	$(u_4;v_6,v_{12},v_{15})$	$(u_5;v_5,v_7,v_{16})$	$(u_6;v_8,v_{14},v_{17})$
$\{0,2,9,11\}$	$(u_0;v_0,v_2,v_{11})$	$(u_1;v_1,v_3,v_{10})$	$(u_3;v_5,v_{12},v_{14})$	$(u_4;v_4,v_6,v_{13})$	$(u_6;v_8,v_{15},v_{17})$	$(u_7;v_7,v_9,v_{16})$
$\{0,3,4,11\}$	$(u_0;v_0,v_4,v_{11})$	$(u_1;v_1,v_5,v_{12})$	$(u_4;v_7,v_8,v_{15})$	$(u_6;v_6,v_{10},v_{17})$	$(u_{10};v_3,v_{13},v_{14})$	$(u_{16};v_2,v_9,v_{16})$
$\{0,3,5,11\}$	$(u_0;v_0,v_3,v_{11})$	$(u_1;v_1,v_6,v_{12})$	$(u_2;v_2,v_5,v_{13})$	$(u_4;v_4,v_9,v_{15})$	$(u_5;v_8,v_{10},v_{16})$	$(u_{14};v_7,v_{14},v_{17})$
$\{0,3,6,11\}$	$(u_0;v_0,v_3,v_6)$	$(u_1;v_1,v_4,v_7)$	$(u_2;v_2,v_5,v_8)$	$(u_9;v_9,v_{12},v_{15})$	$(u_{10};v_{10},v_{13},v_{16})$	$(u_{11};v_{11},v_{14},v_{17})$
$\{0,3,7,11\}$	$(u_0;v_0,v_7,v_{11})$	$(u_1;v_1,v_4,v_8)$	$(u_3;v_3,v_{10},v_{14})$	$(u_6;v_6,v_{13},v_{17})$	$(u_9;v_2,v_9,v_{16})$	$(u_{12};v_5,v_{12},v_{15})$
$\{0,3,8,11\}$	$(u_0;v_0,v_3,v_{11})$	$(u_1;v_1,v_4,v_9)$	$(u_2;v_2,v_5,v_{10})$	$(u_4;v_7,v_{12},v_{15})$	$(u_5;v_8,v_{13},v_{16})$	$(u_6;v_6,v_{14},v_{17})$
$\{0,3,9,11\}$	$(u_0;v_0,v_3,v_9)$	$(u_1;v_1,v_{10},v_{12})$	$(u_2;v_2,v_5,v_{11})$	$(u_4;v_4,v_7,v_{13})$	$(u_5;v_8,v_{14},v_{16})$	$(u_6;v_6,v_{15},v_{17})$
$\{0,3,10,11\}$	$(u_0;v_0,v_3,v_{10})$	$(u_1;v_1,v_4,v_{11})$	$(u_2;v_2,v_5,v_{12})$	$(u_3;v_6,v_{13},v_{14})$	$(u_6;v_9,v_{16},v_{17})$	$(u_{15};v_7,v_8,v_{15})$
$\{0,4,5,11\}$	$(u_0;v_0,v_4,v_{11})$	$(u_1;v_1,v_5,v_6)$	$(u_2;v_2,v_7,v_{13})$	$(u_4;v_8,v_9,v_{15})$	$(u_{10};v_3,v_{10},v_{14})$	$(u_{12};v_{12},v_{16},v_{17})$
$\{0,4,6,11\}$	$(u_0;v_0,v_4,v_6)$	$(u_1;v_1,v_5,v_{12})$	$(u_2;v_2,v_8,v_{13})$	$(u_3;v_3,v_7,v_9)$	$(u_{10};v_{10},v_{14},v_{16})$	$(u_{11};v_{11},v_{15},v_{17})$
$\{0,4,7,11\}$	$(u_0;v_0,v_4,v_7)$	$(u_1;v_1,v_5,v_{12})$	$(u_3;v_3,v_{10},v_{14})$	$(u_4;v_8,v_{11},v_{15})$	$(u_6;v_6,v_{13},v_{17})$	$(u_9;v_2,v_9,v_{16})$
$\{0,4,8,11\}$	$(u_0;v_0,v_4,v_8)$	$(u_1;v_1,v_5,v_9)$	$(u_6;v_6,v_{10},v_{14})$	$(u_7;v_7,v_{11},v_{15})$	$(u_{12};v_2,v_{12},v_{16})$	$(u_{13};v_3,v_{13},v_{17})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,4,9,11\}$	$(u_0;v_0,v_4,v_{11})$	$(u_1;v_1,v_5,v_{12})$	$(u_3;v_3,v_7,v_{14})$	$(u_4;v_8,v_{13},v_{15})$	$(u_6;v_6,v_{10},v_{17})$	$(u_{16};v_2,v_9,v_{16})$
$\{0,4,10,11\}$	$(u_0;v_0,v_{10},v_{11})$	$(u_1;v_1,v_5,v_{12})$	$(u_3;v_3,v_7,v_{14})$	$(u_4;v_4,v_8,v_{15})$	$(u_6;v_6,v_{16},v_{17})$	$(u_9;v_2,v_9,v_{13})$
$\{0,5,6,11\}$	$(u_0;v_0,v_5,v_6)$	$(u_1;v_1,v_7,v_{12})$	$(u_2;v_2,v_8,v_{13})$	$(u_3;v_3,v_9,v_{14})$	$(u_4;v_4,v_{10},v_{15})$	$(u_{11};v_{11},v_{16},v_{17})$
$\{0,5,7,11\}$	$(u_0;v_0,v_5,v_7)$	$(u_1;v_1,v_6,v_{12})$	$(u_2;v_2,v_9,v_{13})$	$(u_3;v_3,v_8,v_{14})$	$(u_{10};v_{10},v_{15},v_{17})$	$(u_{11};v_4,v_{11},v_{16})$
$\{0,5,8,11\}$	$(u_0;v_0,v_5,v_8)$	$(u_1;v_1,v_9,v_{12})$	$(u_2;v_2,v_7,v_{13})$	$(u_6;v_6,v_{14},v_{17})$	$(u_{10};v_3,v_{10},v_{15})$	$(u_{11};v_4,v_{11},v_{16})$
$\{0,5,9,11\}$	$(u_0;v_0,v_5,v_9)$	$(u_1;v_1,v_{10},v_{12})$	$(u_2;v_2,v_7,v_{13})$	$(u_3;v_3,v_8,v_{14})$	$(u_6;v_6,v_{15},v_{17})$	$(u_{11};v_4,v_{11},v_{16})$
$\{0,5,10,11\}$	$(u_0;v_0,v_5,v_{10})$	$(u_1;v_1,v_6,v_{12})$	$(u_2;v_2,v_7,v_{13})$	$(u_4;v_4,v_{14},v_{15})$	$(u_6;v_{11},v_{16},v_{17})$	$(u_{16};v_3,v_8,v_9)$
$\{0,6,7,11\}$	$(u_0;v_0,v_6,v_{11})$	$(u_1;v_1,v_7,v_8)$	$(u_3;v_3,v_{10},v_{14})$	$(u_6;v_{12},v_{13},v_{17})$	$(u_9;v_2,v_9,v_{15})$	$(u_{16};v_4,v_5,v_{16})$
$\{0,6,8,11\}$	$(u_0;v_0,v_8,v_{11})$	$(u_1;v_1,v_7,v_9)$	$(u_2;v_2,v_{10},v_{13})$	$(u_6;v_{12},v_{14},v_{17})$	$(u_{15};v_3,v_5,v_{15})$	$(u_{16};v_4,v_6,v_{16})$
$\{0,6,9,11\}$	$(u_0;v_0,v_6,v_9)$	$(u_1;v_1,v_7,v_{12})$	$(u_4;v_4,v_{10},v_{13})$	$(u_5;v_5,v_{14},v_{16})$	$(u_{11};v_2,v_{11},v_{17})$	$(u_{15};v_3,v_8,v_{15})$
$\{0,6,10,11\}$	$(u_0;v_0,v_6,v_{10})$	$(u_1;v_1,v_7,v_{12})$	$(u_2;v_2,v_8,v_{13})$	$(u_3;v_3,v_9,v_{14})$	$(u_5;v_5,v_{15},v_{16})$	$(u_{11};v_4,v_{11},v_{17})$
$\{0,7,8,11\}$	$(u_0;v_0,v_7,v_8)$	$(u_1;v_1,v_9,v_{12})$	$(u_3;v_3,v_{10},v_{11})$	$(u_6;v_6,v_{13},v_{14})$	$(u_9;v_2,v_{16},v_{17})$	$(u_{15};v_4,v_5,v_{15})$
$\{0,7,9,11\}$	$(u_0;v_0,v_7,v_9)$	$(u_1;v_1,v_8,v_{12})$	$(u_2;v_2,v_{11},v_{13})$	$(u_5;v_5,v_{14},v_{16})$	$(u_{10};v_3,v_{10},v_{17})$	$(u_{15};v_4,v_6,v_{15})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,7,10,11\}$	$(u_0; v_0, v_7, v_{11})$	$(u_1; v_1, v_8, v_{12})$	$(u_4; v_4, v_{14}, v_{15})$	$(u_6; v_6, v_{13}, v_{17})$	$(u_{10}; v_2, v_3, v_{10})$	$(u_{16}; v_5, v_9, v_{16})$
$\{0,1,6,12\}$	$(u_0; v_0, v_1, v_{12})$	$(u_1; v_2, v_7, v_{13})$	$(u_2; v_3, v_8, v_{14})$	$(u_3; v_4, v_9, v_{15})$	$(u_4; v_5, v_{10}, v_{16})$	$(u_5; v_6, v_{11}, v_{17})$
$\{0,1,7,12\}$	$(u_0; v_0, v_1, v_7)$	$(u_1; v_2, v_8, v_{13})$	$(u_2; v_3, v_9, v_{14})$	$(u_3; v_4, v_{10}, v_{15})$	$(u_4; v_5, v_{11}, v_{16})$	$(u_5; v_6, v_{12}, v_{17})$
$\{0,2,6,12\}$	Two-component graph see $n = 9$ and $D = \{0,1,3,6\}$					
$\{0,2,7,12\}$	$(u_0; v_0, v_2, v_7)$	$(u_1; v_1, v_8, v_{13})$	$(u_2; v_4, v_9, v_{14})$	$(u_3; v_3, v_{10}, v_{15})$	$(u_4; v_6, v_{11}, v_{16})$	$(u_5; v_5, v_{12}, v_{17})$
$\{0,2,8,12\}$	Two-component graph see $n = 9$ and $D = \{0,1,4,6\}$					
$\{0,3,6,12\}$	Three-component graph see $n = 6$ and $D = \{0,1,2,4\}$					
$\{0,3,7,12\}$	$(u_0; v_0, v_3, v_{12})$	$(u_1; v_1, v_4, v_{13})$	$(u_2; v_2, v_9, v_{14})$	$(u_3; v_6, v_{10}, v_{15})$	$(u_4; v_7, v_{11}, v_{16})$	$(u_5; v_5, v_8, v_{17})$
$\{0,3,8,12\}$	$(u_0; v_0, v_3, v_8)$	$(u_1; v_1, v_4, v_9)$	$(u_2; v_2, v_{10}, v_{14})$	$(u_3; v_6, v_{11}, v_{15})$	$(u_4; v_7, v_{12}, v_{16})$	$(u_5; v_5, v_{13}, v_{17})$
$\{0,3,9,12\}$	Three-component graph see $n = 6$ and $D = \{0,1,3,4\}$					
$\{0,4,6,12\}$	Two-component graph see $n = 9$ and $D = \{0,2,3,6\}$					
$\{0,4,7,12\}$	$(u_0; v_0, v_4, v_7)$	$(u_1; v_1, v_5, v_{13})$	$(u_2; v_2, v_6, v_{14})$	$(u_3; v_3, v_{10}, v_{15})$	$(u_4; v_8, v_{11}, v_{16})$	$(u_5; v_9, v_{12}, v_{17})$
$\{0,4,8,12\}$	Two-component graph see $n = 9$ and $D = \{0,2,4,6\}$					

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,4,9,12\}$	$(u_0;v_0,v_4,v_{12})$	$(u_1;v_1,v_5,v_{10})$	$(u_2;v_2,v_6,v_{11})$	$(u_3;v_3,v_7,v_{15})$	$(u_4;v_8,v_{13},v_{16})$	$(u_5;v_9,v_{14},v_{17})$
$\{0,4,10,12\}$	Two-component graph see $n=9$ and $D=\{0,2,5,6\}$					
$\{0,5,6,12\}$	$(u_0;v_0,v_5,v_{12})$	$(u_1;v_1,v_6,v_{13})$	$(u_2;v_2,v_7,v_{14})$	$(u_3;v_3,v_8,v_{15})$	$(u_4;v_4,v_9,v_{16})$	$(u_5;v_{10},v_{11},v_{17})$
$\{0,5,7,12\}$	$(u_0;v_0,v_5,v_7)$	$(u_1;v_1,v_6,v_{13})$	$(u_2;v_2,v_9,v_{14})$	$(u_3;v_3,v_8,v_{15})$	$(u_4;v_4,v_{11},v_{16})$	$(u_5;v_{10},v_{12},v_{17})$
$\{0,5,8,12\}$	$(u_0;v_0,v_5,v_8)$	$(u_1;v_1,v_6,v_9)$	$(u_2;v_2,v_7,v_{14})$	$(u_3;v_3,v_{11},v_{15})$	$(u_4;v_4,v_{12},v_{16})$	$(u_5;v_{10},v_{13},v_{17})$
$\{0,5,9,12\}$	$(u_0;v_0,v_5,v_{12})$	$(u_1;v_1,v_6,v_{13})$	$(u_2;v_2,v_7,v_{11})$	$(u_3;v_3,v_8,v_{15})$	$(u_4;v_4,v_9,v_{16})$	$(u_5;v_{10},v_{14},v_{17})$
$\{0,5,10,12\}$	$(u_0;v_0,v_5,v_{10})$	$(u_1;v_6,v_{11},v_{13})$	$(u_2;v_2,v_7,v_{14})$	$(u_3;v_3,v_8,v_{15})$	$(u_4;v_4,v_9,v_{16})$	$(u_7;v_1,v_{12},v_{17})$
$\{0,5,11,12\}$	$(u_0;v_0,v_5,v_{11})$	$(u_1;v_1,v_6,v_{12})$	$(u_2;v_2,v_7,v_{13})$	$(u_3;v_3,v_8,v_{14})$	$(u_4;v_4,v_9,v_{15})$	$(u_5;v_{10},v_{16},v_{17})$
$\{0,6,7,12\}$	$(u_0;v_0,v_6,v_7)$	$(u_1;v_1,v_8,v_{13})$	$(u_2;v_2,v_9,v_{14})$	$(u_3;v_3,v_{10},v_{15})$	$(u_4;v_4,v_{11},v_{16})$	$(u_5;v_5,v_{12},v_{17})$
$\{0,6,8,12\}$	Two-component graph see $n=9$ and $D=\{0,3,4,6\}$					
$\{0,6,9,12\}$	Three-component graph see $n=6$ and $D=\{0,2,3,4\}$					
$\{0,6,10,12\}$	Two-component graph see $n=9$ and $D=\{0,3,5,6\}$					
$\{0,6,11,12\}$	$(u_0;v_0,v_6,v_{11})$	$(u_1;v_1,v_7,v_{12})$	$(u_2;v_2,v_8,v_{13})$	$(u_3;v_3,v_9,v_{14})$	$(u_4;v_4,v_{10},v_{15})$	$(u_5;v_5,v_{16},v_{17})$

Table 6.5 – *Continued on next page*

Table 6.5 – *Continued from previous page*

Generator Set	Star 1	Star 2	Star 3	Star 4	Star 5	Star 6
$\{0,3,8,13\}$	$(u_0;v_0,v_3,v_8)$	$(u_1;v_1,v_9,v_{14})$	$(u_3;v_6,v_{11},v_{16})$	$(u_4;v_4,v_{12},v_{17})$	$(u_7;v_2,v_7,v_{15})$	$(u_{10};v_5,v_{10},v_{13})$
$\{0,4,8,13\}$	$(u_0;v_0,v_4,v_{13})$	$(u_1;v_5,v_9,v_{14})$	$(u_2;v_2,v_6,v_{10})$	$(u_3;v_3,v_7,v_{16})$	$(u_4;v_8,v_{12},v_{17})$	$(u_{11};v_1,v_{11},v_{15})$
$\{0,4,9,13\}$	$(u_0;v_0,v_4,v_9)$	$(u_1;v_1,v_5,v_{14})$	$(u_3;v_3,v_7,v_{12})$	$(u_4;v_8,v_{13},v_{17})$	$(u_6;v_6,v_{10},v_{15})$	$(u_7;v_2,v_{11},v_{16})$
$\{0,5,8,13\}$	$(u_0;v_0,v_8,v_{13})$	$(u_1;v_1,v_6,v_{14})$	$(u_3;v_3,v_{11},v_{16})$	$(u_4;v_4,v_9,v_{17})$	$(u_7;v_2,v_7,v_{12})$	$(u_{10};v_5,v_{10},v_{15})$
$\{0,5,9,13\}$	$(u_0;v_0,v_5,v_9)$	$(u_1;v_1,v_{10},v_{14})$	$(u_2;v_2,v_7,v_{15})$	$(u_3;v_3,v_8,v_{12})$	$(u_4;v_4,v_{13},v_{17})$	$(u_{11};v_6,v_{11},v_{16})$
$\{0,5,10,13\}$	$(u_0;v_0,v_5,v_{13})$	$(u_1;v_1,v_6,v_{11})$	$(u_3;v_3,v_8,v_{16})$	$(u_4;v_4,v_9,v_{14})$	$(u_7;v_7,v_{12},v_{17})$	$(u_{10};v_2,v_{10},v_{15})$

Table 6.5: S_3-cover of Partite Set V for $n = 18$

References

[1] N. Alon. A note on the decomposition of graphs into isomorphic matchings. *Acta Mathematica Hungarica*, 42(3-4):221–223, 1983.

[2] N. Alon and J. Spencer. *Probabilistic Method*. John Wiley & Sons, 1991.

[3] L. W. Beineke and R. J. Wilson. *Topics in Algebraic Graph Theory*. Cambridge University Press, 2004.

[4] A. Bockmayr and K. Reinert. Discrete math for bioinformatics. Retrieved from: `http://www.mi.fu-berlin.de/en/inf/groups/abi/teaching/lectures_past/WS1112/V____Discrete_Mathematics_for_Bioinformatics__P1/index.html`, November 2010.

[5] J. Bondy and U. Murty. *Graph Theory with Application*. Elsevier Science Ltd/North-Holland, Nowhere, 1974.

[6] M. A. Boutiche. Control of some graph invariants in dynamic routing. *Modelling, Computation and Optimization in Information Systems and Management Sciences Communications in Computer and Information Science*, 14:52–58, 2008.

[7] N. Bray. Graph lexicographic product. `http://mathworld.wolfram.com/GraphLexicographicProduct.html`.

[8] P. Cain. Decomposition of complete graphs into star. *Bulletin of the Australian Mathematical Society*, 10:23–30, 1974.

[9] N. Cavenagh. Decomposition of complete tripartite graphs into k-cycles. *Australasian Journal of Combinatorics*, 18:193–200, 1998.

[10] C. Chekuri, S. Khanna, and F. B. Shepherd. The all-or-nothing multicommodity flow problem. *SIAM Journal on Computing*, 42:1467–1493, 2013.

[11] S. Cichacz, D. Froncek, and P. Kovar. Decomposition of complete bipartite graphs into generalized prisms. *Decomposition of complete bipartite graphs into Generalized prisms*, 34(1):104–110, 2012.

[12] E. Cohen and M. Tarsi. NP completeness of graph decomposition problems. *Journal of Complexity*, 7(2):200–212, June 1991.

[13] C. Colbourn and J. Dinitz. *Handbook of Combinatorial Designs, Second Edition*. Discrete Mathematics and Its Applications. CRC Press, 2010.

[14] I. Csiszar and J. Korner. Graph decomposition: A new key to coding theorems. *Information Theory, IEEE Transactions on*, 27(1):5–12, January 1981.

[15] J. Deuerlein, A. Wolters, L. Meyer-Harries, and A. R. Simpson. *Graph Decomposition in Risk Analysis and Sensor Placement for Water Distribution Network Security*, chapter 37, pages 394–411. American Society of Civil Engineers, 2010.

[16] D. Dor and M. Tarsi. Graph decomposition is NP-complete, a complete proof of Holyer's conjecture. In *Proc. 20th ACM STOC, ACM Press*, pages 252–263. Press, 1995.

[17] D. Dor and M. Tarsi. Graph decomposition is NP-complete: A complete proof of Holyer's conjecture. *SIAM Journal on Computing*, 26(4):1166–1187, 1997.

[18] P. Erdös. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[19] P. Erdös, A. Goodman, and L. Posa. The representation of a graph by set intersection. *Canadian Journal of Mathematics*, 18:106–112, 1966.

[20] O. Favaron, Z. Lonc, and M. Truszczynski. Decompositions of graphs into graphs with three edges. *Ars Combinatoria*, 20:125–146, 1984.

[21] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[22] I. S. Hamid and M. Joseph. A variation of decomposition under a length constraint. *International J. Math. Combin. Vol*, 4:35–45, 2011.

[23] D. G. Hoffman. The real truth about star designs. *Discrete Mathematics*, 284(1-3), July 2004.

[24] I. Holyer. *The Computational Complexity of Graph Theory Problems.* Doctoral dissertation, University of Cambridge, 1981.

[25] W. Imrich, S. Klavzar, and D. F. Rall. *Graphs and their Cartesian Product.* A.K. Peters, 2008.

[26] M. S. Jacobson, M. Truszczyński, and Z. Tuza. Decompositions of regular bipartite graphs. *Discrete mathematics*, 89(1):17–27, 1991.

[27] J. Kleinberg and E. Tardos. *Algorithm Design.* Addison-Wesley, 2005.

[28] H.-C. Lee and J.-J. Lin. Decomposition of the complete bipartite graph with a 1-factor removed into cycles and stars. *Discrete Mathematics*, 313(20):2354–2358, 2013.

[29] Z. Lonc. Towards a solution of the holyer's problem. In J. van Leeuwen, editor, *Graph-Theoretic Concepts in Computer Science*, volume 790 of *Lecture Notes in Computer Science*, pages 144–152. Springer Berlin Heidelberg, 1994.

[30] I. Mandoiu and A. Zelikovsky. *Bioinformatics Algorithms: Techniques and Applications.* John Wiley & Sons, 2010.

[31] M. Priesler and M. Tarsi. On the decomposition of graphs into copies P_3tK_2. *Ars Combinatoria*, 35:325–333, 1993.

[32] S. Severini and E. W. Weisstein. Crown graph. `http://mathworld.wolfram.com/CrownGraph.html`.

[33] T. W. Shyu. *The Decomposition of Complete Graphs, Complete Bipartite Graphs and Crowns.* Doctoral dissertation, National Central University, Taiwan, 1996.

[34] S. Skiena. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica.* Addison-Wesley, 1990. pp. 128 and 135-139.

[35] D. Sotteau. Decompositions of $K_{m,n}$ ($K_{m,n}*$) into cycles (circuits) of length $2k$. *Journal of Combinatorial Theory, Series B*, 30(1):75–81, February 1981.

[36] T. Sousa. The H-decomposition problem for graphs. *Applied Mathematics*, 3(11):1719, 2012.

[37] K. Ushio, S. Tazawa, and S. Yamamoto. On claw-decomposition of a complete multipartite graph. *Hiroshima Math. Journal*, 8(1):207–2–1, 1978.

[38] S. Yamamoto, H. Ikeda, S. Shige-eda, K. Ushio, and N. Hamada. On claw-decomposition of complete graphs and complete bigraphs. *Hiroshima Mathematical Journal*, 5(1):33–42, 1975.

[39] R. Yuster. Tree decomposition of graphs. *Random Structures and Algorithms*, 12(3):237–251, 1998.