The Theory and Applications of Stratified Graphs

Reza Rashidi

Western Michigan University

Follow this and additional works at: https://scholarworks.wmich.edu/dissertations

Part of the Applied Mathematics Commons

Recommended Citation
https://scholarworks.wmich.edu/dissertations/1859

This Dissertation - Open Access is brought to you for free and open access by the Graduate College at ScholarWorks at WMU. It has been accepted for inclusion in Dissertations by an authorized administrator of ScholarWorks at WMU. For more information, please contact maira.bundza@wmich.edu.
THE THEORY AND APPLICATIONS OF STRATIFIED GRAPHS

by

Reza Rashidi

A Dissertation
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Doctor of Philosophy
Department of Mathematics and Statistics

Western Michigan University
Kalamazoo, Michigan
December 1994
Physical design is one of several stages in the design of a VLSI chip. In this stage, the specifications of an electrical circuit are converted into a geometrical model. Problems concerning the physical design stage can often be studied by means of graphs. The problems encountered here are routing problems and concern placement of vertices, which represent wires, into layers. All this gives rise to a class of graphs whose vertices are partitioned into classes. Such graphs are called stratified graphs. In this dissertation, we formally define stratified graphs, study their properties, and investigate various algorithmic problems related to these graphs.

In Chapter I, the concept of stratified graphs is formally defined and basic terminology and notation are introduced. Chapter II concerns degrees and degree multisets in stratified graphs. The concept of color pattern is defined. Each vertex of a stratified graph has, in fact, a unique color pattern. Those multisets of color patterns, called arrangements, that are realized in a stratified graph are characterized. Color-regular graphs are defined and, with small restriction, it is shown that color-regular 2-stratified graphs exist with prescribed color patterns, and, in fact, the minimum order of such graphs is determined. It is shown that the so-called stratification problem that arises in this context is NP-complete.

In Chapter III, the emphasis is on distance in stratified graphs. The concepts of eccentricity, radius, diameter, center, and periphery are defined for these
graphs and investigated, as well as other distance-related concepts. It is shown that for every set \( S = \{H_1, H_2, \ldots, H_k\} \) of \( k \)-stratified graphs, there exists a \( k \)-stratified graph \( G \) whose \( k \) centers are precisely the graphs in \( S \). Those \( k \)-stratified graphs that are peripheries of \( k \)-stratified graphs are characterized. The distance-related concepts of proximity and seclusion, for which there is no analogue in ordinary graphs, is introduced and studied.

Chapter IV deals with a variety of algorithmic problems related to stratified graphs. A dynamic programming algorithm is presented for determining a smallest color-spanning subtree. In Chapter V, specific routing problems from the VLSI physical design stage are modeled by stratified graphs and related algorithms are presented. The dissertation concludes in Chapter VI with a discussion of open problems and some directions for future research.
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
To Sohyla, Nima, and Leila
for their love, patience and understanding.
ACKNOWLEDGEMENTS

I thank Professor Naveed Sherwani for his guidance and support, particularly in the early stages of this project. I am extremely grateful to Professor Gary Chartrand for helping me in every step of the way, for the duration of this research, and especially for his patience, guidance and support during the writing of this dissertation. I also wish to thank Professor Yousef Alavi, Professor Dionysios Kountanis, and Professor Dennis Pence for serving on my committee and for their support and encouragement throughout the years.

I also wish to acknowledge the support of Dr. Harley Behm, director of the University Computing Services and my supervisor at Western Michigan University. Without his support and understanding, I would not have been able to finish this project. Many thanks also go to my friends and colleagues Bruce P. Mull, Xiaoxia He, Matthew Shafer, Bardia Madani, Julie Scott, Bruce Paananen, and Margie Easter for their direct and indirect help particularly in the last months of writing this dissertation.

I am deeply grateful to my father who has emphasized the importance of education throughout my life and has supported me throughout this endeavor with generous financial support, my mother who has been a continuous source of pride in my life, and my sister for her friendship. Finally, my special gratitude goes to Sousan Tavallai for exceptional inspiration and friendship that she afforded me when I needed it the most.

Reza Rashidi
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ................................................................. ii
LIST OF TABLES .......................................................................... v
LIST OF FIGURES .......................................................................... vi

CHAPTER

I. PRELIMINARIES ................................................................. 1
  1.1 Introduction ............................................................... 1
  1.2 Definitions and Notations .......................................... 4

II. DEGREE IN STRATIFIED GRAPHS ........................................ 13
  2.1 Introduction .............................................................. 13
  2.2 Arrangements .............................................................. 14
  2.3 Color-Regular Stratified Graphs ................................. 21
  2.4 Degree-Vector Multisets ............................................ 27

III. DISTANCE IN STRATIFIED GRAPHS ................................. 39
  3.1 Introduction .............................................................. 39
  3.2 Eccentricity, Radius, and Diameter ......................... 40
  3.3 Centers in Stratified Graphs ................................. 45
  3.4 Centers in Properly 2-Colored 2-Stratified Trees 52
  3.5 Periphery ................................................................. 59
  3.6 Proximity, Maximum Proximity, Seclusion .............. 61

IV. ALGORITHMS IN STRATIFIED GRAPHS .............................. 64
  4.1 Introduction .............................................................. 64
  4.2 Finding Optimized Subgraphs ................................. 64
  4.3 Color-Spanning Subgraphs ....................................... 72
# Table of Contents—Continued

## CHAPTER

4.3.1 Color-Spanning Cliques .......................... 72
4.3.2 Color-Spanning Paths ............................... 73
4.3.3 Color-Spanning Trees ................................. 76
4.4 Domination and Independent Sets ...................... 85
4.5 Domination in 2-Stratified Trees ...................... 85
4.6 Independence in Stratified Graphs ..................... 87

## V. APPROXIMATION ALGORITHMS FOR STRATIFIED GRAPHS WITH APPLICATIONS ............................................................... 91

5.1 Introduction .............................................................. 91
5.2 Cell Model and Preliminaries .............................. 93
5.2.1 OTC Algorithm Overview ............................. 94
5.2.2 Vacant Terminals and Net Classification ........... 95
5.3 Existing Algorithms for the Net Selection ............. 97
5.4 Improved Algorithm for the Two Layer Process ...... 99
5.5 Approximation Algorithm for Three Layer Process . 102
5.6 Experimental Results ............................................... 103
5.7 Conclusions .............................................................. 104

## VI. OPEN PROBLEMS ......................................................... 107

BIBLIOGRAPHY ................................................................. 116

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
LIST OF TABLES

1. A Summary of Properties of the 3-stratified Graph of Figure 2 . . 5
2. Degrees of the Vertices of the 3-stratified Graph of Figure 2 . . . 6
3. Application of Dijkstra's Algorithm to Digraph in Figure 36 . . . 66
4. The Table Computed by MinColorSpSubtree for a 3-stratified Tree 78
5. Experimental Results - PRIMARY 1 ................................. 100
6. Deutsch Difficult Example ................................. 100

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
# LIST OF FIGURES

1. A 3 Layer VHV Routing Model ................................................... 2
2. A 3-stratified Graph. ................................................................. 3
3. Two Isomorphic 3-stratified Graphs. ......................................... 7
4. There are 89 Stratified Graphs on 4 Vertices. ......................... 9
5. A Self-complementary 2-stratified Graph. ............................... 12
6. Two Nonisomorphic Stratified Graphs With the Same Arrangement. 14
7. A 2-stratified Graph Corresponding to the Arrangement $\mathcal{A}$. 16
8. Graphs With Degree Sequences $\sigma_R$ and $\sigma_B$. .................. 20
9. A Bipartite Graph With Bigraphical Sequences $\sigma_{R,B}$ and $\sigma_{B,R}$. 20
10. A 2-stratified Graph With Arrangement $\mathcal{A}$. ..................... 21
11. A Color-regular 2-stratified Graph. ........................................ 22
12. Construction of a Color-regular Graph With Color Patterns $(R;9,6)$ and $(B;4,8)$. ................................................................. 24
13. Construction of a Color-regular Graph With Color Patterns $(R;2,2,2)$, $(B;3,2,3)$, $(G;5,5,2)$. ......................................................... 26
14. Stratified Graphs With the Same Degree-vector Multiset but Having Distinct Density Vectors. .............................................. 27
15. A Sequential Unicolor-block Listing: RRRBRRBB. ..................... 33
16. Case (I), Alternating Colors: RBRB...RB. ............................... 35
17. Case (II), $m = 2, a = 5, b = 3$: RRRBRBRBRBR. .................... 36
18. Case (III), Alternating Pairs: RRBB...RRBB. ......................... 36
19. Case 3, $a \neq 0$, or $b \neq 0$: BR...RB...B...RRBBRR. ......... 37
20. Red Eccentricity in a 2-stratified Graph. ............................... 39
21. A 5-stratified Graph With X-radius 4 and X-diameter 7. .......... 40
List of Figures—Continued

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.</td>
<td>A 9-stratified Graph With X-radius 3 and X-diameter 8.</td>
<td>41</td>
</tr>
<tr>
<td>23.</td>
<td>A Stratified Graph Having Only One Vertex of a Given X-eccentricity.</td>
<td>42</td>
</tr>
<tr>
<td>24.</td>
<td>Three 2-stratified Graphs and Their Red Centers.</td>
<td>44</td>
</tr>
<tr>
<td>25.</td>
<td>$C_R(G)$ and $C_B(G)$ Are Disconnected. $(C_R(G) \cup C_B(G)) = C(G)$.</td>
<td>45</td>
</tr>
<tr>
<td>26.</td>
<td>$(C_R(G) \cap C_B(G)) = C(G)$.</td>
<td>46</td>
</tr>
<tr>
<td>27.</td>
<td>A $k$-stratified Graph $G$ With X-center $C_X(G) = H$.</td>
<td>46</td>
</tr>
<tr>
<td>28.</td>
<td>Construction of a $k$-stratified Graph $G$ With $k$ Prescribed Centers.</td>
<td>47</td>
</tr>
<tr>
<td>29.</td>
<td>A 3-stratified Graph $G$ With Centers at Distance 4.</td>
<td>49</td>
</tr>
<tr>
<td>30.</td>
<td>The Set of Central Vertices of $G$ Is Not a Subset of the Union of the Red and Blue Centers.</td>
<td>54</td>
</tr>
<tr>
<td>31.</td>
<td>A 2-stratified Graph $G$ and Its Red Periphery.</td>
<td>56</td>
</tr>
<tr>
<td>32.</td>
<td>A 2-stratified Graph $G$ and Its Red Seclusion.</td>
<td>59</td>
</tr>
<tr>
<td>33.</td>
<td>A 2-stratified Path With $\Delta_B(G) = 5$ and $\Delta_R(G) = 8$.</td>
<td>59</td>
</tr>
<tr>
<td>34.</td>
<td>The $l$-stratified Subgraph $H$ is the $X$-seclusion of the $k$-stratified Graph $G$.</td>
<td>60</td>
</tr>
<tr>
<td>35.</td>
<td>Algorithm Find.Unicolored.</td>
<td>63</td>
</tr>
<tr>
<td>36.</td>
<td>A 2-stratified Graph $G$ With Corresponding Weighted Digraph $D$.</td>
<td>65</td>
</tr>
<tr>
<td>37.</td>
<td>Algorithm Find.Rainbow.Clique.</td>
<td>69</td>
</tr>
<tr>
<td>38.</td>
<td>Algorithm Find.Rainbow.SubPath.</td>
<td>70</td>
</tr>
<tr>
<td>40.</td>
<td>A 3-stratified Tree.</td>
<td>75</td>
</tr>
<tr>
<td>41.</td>
<td>Algorithm MinColorSpSubtree.</td>
<td>79</td>
</tr>
<tr>
<td>42.</td>
<td>A 3-stratified Graph.</td>
<td>80</td>
</tr>
<tr>
<td>43.</td>
<td>Domination in a 2-stratified Graph.</td>
<td>80</td>
</tr>
<tr>
<td>44.</td>
<td>Algorithm Find.Min.Dominating($F$).</td>
<td>82</td>
</tr>
</tbody>
</table>
45. A 2-stratified Graph Satisfying Necessary Conditions for LAP-2. 83
46. Standard Cell Model. ......................................................... 89
47. Three Basic Net Types. ....................................................... 92
48. Algorithm 2OTC. ............................................................... 96
49. Performance of Algorithm 2OTC. ......................................... 97
50. Algorithm 3OTC. ............................................................... 98
51. Three Layer Approximation. .............................................. 99
52. A 4-stratified Tree With No Color-cut Vertex. ...................... 103
53. The Total Proximity of \( v \) is Greater Than the Minimum Total Proximity. ......................................................... 105
54. The Vertex \( v \) With Minimum Total Proximity Does Not Belong to the Minimum Color-spanning Subtree. ......................................................... 106
CHAPTER I

PRELIMINARIES

1.1 Introduction

Graphs are often useful mathematical models for structures and relationships that occur in real-life phenomena. Design of a Very Large Scale Integrated Circuit (VLSI) chip involves many complex processes. Currently, a typical VLSI chip consists of millions of transistors assembled through layering of various material in a silicon base. At some point during this process the designer of an integrated circuit (IC) transforms a circuit description into a geometric description, which is known as a layout. The process of converting the specifications of an electrical circuit into a layout is called the physical design. Due to the large number of components and the exacting details required, the physical design is a very complex process that is not practical without automation. VLSI design automation is the study of algorithms and data structures employed in the physical design process [N.A93].

Many of the problems encountered in the physical design process are modeled by graphs. Various routing problems [N.A88, NJ89] and via minimization problems [HKC89, MT89] are among such problems. In recent years, advances in VLSI fabrication technology have made it possible to use more than two routing layers for interconnection. In fact, the two most popular processors in the market today, the PowerPC chip designed by Motorola, IBM, and Apple, as well as the Pentium processor designed and manufactured by the Intel Corporation, use three or more layers. Figure 1 depicts a 3 layer vertical-horizontal-vertical (VHV)

1

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
routing problem in a standard cell architecture. In the design of algorithms to solve the multilayer routing problems encountered in this process, it is desirable to use graphs in which the vertices are partitioned into classes.

![Diagram of a 3 Layer VHV Routing Model](image)

Figure 1. A 3 Layer VHV Routing Model.

Dividing the vertex set of a graph into classes according to some prescribed rule is also a fundamental process in graph theory. The vertices of a graph can be divided into cut-vertices and non-cut-vertices. Equivalently, the vertices of a tree can be divided into its leaves and non-leaves. The vertex set of a graph is partitioned according to the degrees of its vertices. When studying distance, the vertices of a connected graph are partitioned according to their eccentricities. Also, when a vertex (root) is chosen, the vertices are divided into classes according to their distance from the root. Probably the best known example of this process is graph coloring, where the vertex set of a graph is partitioned into classes each of which is independent in the graph.

Motivated by these observations, we define a graph to be a *stratified graph* if its vertex set is partitioned into classes. The primary purpose of this dissertation
is to study the theory and applications of these structures.

In the remainder of Chapter I we introduce the basic terminology used in this study. Chapter II defines concepts that parallel notions of degree and degree sequences in graphs, and addresses the questions of existence and realizability of stratified graphs with prescribed requirements. In Chapter III we present a study of concepts related to distance as it pertains to stratified graphs. In Chapter IV we investigate various algorithmic problems related to these structures. In Chapter V we use stratified graphs to model specific routing problems in the VLSI physical design phase and give approximation algorithms solving them. Chapter VI describes some open problems and discusses possible directions for future research.

1.2 Definitions and Notations

A graph $G$ whose vertex set is partitioned into subsets $S_1, S_2, ..., S_k$ ($k \geq 1$) is referred to as a stratified (or, in this case, a $k$-stratified) graph. Each subset $S_i$ ($1 \leq i \leq k$) is called a stratum and the set $S = \{S_1, S_2, ..., S_k\}$ is called a stratification of $G$. Clearly, the union of the strata of $G$ is $V(G)$.

A 3-stratified graph $G = (S, E)$ is shown in Figure 2, where $S = \{S_1, S_2, S_3\}$ with $S_1 = \{s, t\}$, $S_2 = \{u, v\}$, and $S_3 = \{x, y, z\}$.

![Figure 2. A 3-stratified Graph.](image-url)
We refer to the vertices that belong to the same stratum as having the same color. Thus, the vertices of a $k$-stratified graph $G$ belong to $k$ distinct color classes. The set $\{X_1, X_2, \ldots, X_k\}$ of colors assigned to the vertices form the color spectrum of $G$. We refer to $k$ as the color spectra number of $G$. The number of vertices of color $X_i$ is referred to as the $X_i$-color order of $G$. (For example, if $X_i$ is red, we call this quantity the red order of $G$.) Let $n_{X_i}$ denote the $X_i$-color order of $G$. We refer to $\vec{N} = (n_{X_1}, n_{X_2}, \ldots, n_{X_k})$ as the color density vector of $G$.

Each vertex of a $k$-stratified graph $G$ may be adjacent to vertices from several different color classes or strata. This induces a natural partition of the degree of each vertex such that each element is the number of vertices of a color in the neighborhood of this vertex. We refer to each element of this partition as a color-degree of that vertex. Let $V(G) = \{v_1, v_2, \ldots, v_n\}$. Suppose that each vertex $v_i$ is adjacent to $d^j_i$ vertices from the $j$th stratum ($1 \leq j \leq k$). The color-degree vector of $v_i$ is defined as the $k$-vector of nonnegative integers $(d^1_i, d^2_i, \ldots, d^k_i)$, where then $d^1_i + d^2_i + \ldots + d^k_i = \deg_G(v_i)$. Since the color-degree vector of $v_i$ does not reveal the color of $v_i$, we define yet another vector called the color pattern of a vertex. More formally, we define a color pattern as a vector of the type $P: (X; d_1, d_2, \ldots, d_k)$, where $X$ is a color (an element of some finite predetermined set), and $d_i$ ($1 \leq i \leq k$) is a nonnegative integer. Hence, the color pattern of $v_i$ is the $(k + 1)$-vector $P(v_i) = (\text{color}(v_i); d^1_i, d^2_i, \ldots, d^k_i)$, where $\text{color}(v_i) = X_j$ if and only if $v_i \in S_j$ for $1 \leq j \leq k$.

The arrangement $A(G)$ of a stratified graph $G$ is the multiset of color patterns of the vertices of $G$. The degree-vector multiset $dvs(G)$ of a stratified graph $G$ is the multiset of color-degree vectors of the vertices of $G$. For the stratified graph $G$ of Figure 2 $A(G) = \{(X_3; 0, 0, 2), (X_3; 1, 1, 1), (X_2; 0, 1, 2), (X_3; 2, 1, 1), (X_2; 2, 1, 0), (X_1; 0, 1, 1), (X_1; 0, 1, 2)\}$ and the degree-vector sequence
$d_{vs}(G) = \{(0,0,2), (1,1,1), (0,1,2), (2,1,1), (2,1,0), (0,1,1), (0,1,2)\}$.

A natural problem is to determine whether an arbitrary multiset of color patterns is an arrangement. In other words, is it possible to construct a stratified graph with a given multiset of color patterns as its arrangement? In the next chapter we attempt to answer this and other similar questions.

The graph $G$ of Figure 2 has color spectrum $\{X_1, X_2, X_3\}$, where $X_i$ is the color assigned to the vertices in stratum $S_i$ ($i = 1, 2, 3$). When the number of colors in the spectrum is small, we use names of actual colors to refer to them. In general, we use red, blue, and green for $X_1$, $X_2$, and $X_3$, respectively. Thus the color spectrum of the stratified graph $G$ in Figure 2 is $\{\text{Red, Blue, Green}\}$, or $\{R, B, G\}$ for short. Thus the color spectra number of $G$ is 3. Table 1 summarizes the information regarding the 3-stratified graph $G$ of Figure 2. The subgraph of

<table>
<thead>
<tr>
<th>Strata</th>
<th>$S_1 = {s,t}$</th>
<th>$S_2 = {u,v}$</th>
<th>$S_3 = {x,y,z}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colors</td>
<td>$X_1$ or Red</td>
<td>$X_2$ or Blue</td>
<td>$X_3$ or Green</td>
</tr>
<tr>
<td>Color density vector</td>
<td>Red order 2</td>
<td>Blue order 2</td>
<td>Green order 3</td>
</tr>
</tbody>
</table>

a $k$-stratified graph $G$ induced by the $X_j$-colored vertices ($1 \leq j \leq k$) of $G$ is called the $X_j$-colored subgraph of $G$, which is denoted by $\langle S_j \rangle$. For example, the blue subgraph of $G$ from Figure 2 is $\langle S_2 \rangle \cong P_2$ induced by the vertices $u$ and $v$.

Each edge of $G$ is incident to two vertices of the same color or of different colors. Edges of the latter type are referred to as bicolored edges as opposed to those of the first type, which are called unicolored edges. We denote the number of unicolored and bicolored edges incident to the vertex $v$ by $d_u(v)$ and $d_b(v)$.
respectively. Also let $D_u = \frac{1}{2} \sum_{i=1}^{n} d_u(v_i)$, and $D_b = \frac{1}{2} \sum_{i=1}^{n} d_b(v_i)$. Obviously, $D_u + D_b = m$, that is, the number of edges in $G$. Given the color pattern of the vertex $v$ in $G$, it follows that $d_u(v) = x_j$, where $X_j = \text{color}(v)$ and $d_b(v) = \deg_G(v) - d_u(v)$. Therefore, if we are given the arrangement for the vertices of $G$, it is easy to determine the total number of unicolored and bicolored edges in $G$.

Another problem is whether, for a given degree-vector multiset for $G$, it is possible to determine $d_u$ and $d_b$ for each vertex of $G$. A simpler problem is to determine the total number $D_u$ of unicolored edges in $G$. For the 3-stratified graph $G$ of Figure 2, this information is summarized in Table 2.

**Table 2**

Degrees of the Vertices of the 3-stratified Graph of Figure 2

<table>
<thead>
<tr>
<th>Vertex</th>
<th>$d_u$</th>
<th>$d_b$</th>
<th>$\deg$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>$y$</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$u$</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$z$</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$v$</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$s$</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$t$</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td><strong>6</strong></td>
<td><strong>14</strong></td>
<td><strong>20</strong></td>
</tr>
</tbody>
</table>

For a stratified graph $G$, the underlying graph $G'$ of $G$ is that graph with $V(G') = V(G)$ and $E(G') = E(G)$. If the vertices of $G$ belong to a single color class (or equivalently have no color) or if each color class consists of a single vertex, then $G$ is essentially an ordinary graph. In this sense, stratified graphs serve as a generalization of graphs. Indeed, a 1-stratified graph is an ordinary graph, as is an $n$-stratified graph of order $n$. Unless stated otherwise, whenever we refer
to $G$ as a stratified graph, we shall assume that the vertices of $G$ are partitioned according to some apriori rule into two or more strata.

Next, we describe those conditions under which two stratified graphs are considered to be the same. Two $k$-stratified graphs $G_1$ and $G_2$ are isomorphic (as $k$-stratified graphs), written $G_1 \cong_k G_2$, if there exists a one-to-one correspondence $\phi : V(G_1) \to V(G_2)$ such that (a) $\phi$ preserves adjacency, that is, $uv \in E(G_1)$ if and only if $\phi(u)\phi(v) \in E(G_2)$, and (b) $\phi$ preserves stratification, that is, $u$ and $v$ have the same color in $G_1$ if and only if $\phi(u)$ and $\phi(v)$ have the same color in $G_2$.

For example, the 3-stratified graphs $G_1$ and $G_2$ of Figure 3 are isomorphic, with associated function $\phi$ given below

$$\phi : \begin{align*}
v_1 &\mapsto u_3 \\
v_2 &\mapsto u_2 \\
v_3 &\mapsto u_4 \\
v_4 &\mapsto u_1 \\
v_5 &\mapsto u_5
\end{align*}$$

![Figure 3. Two Isomorphic 3-stratified Graphs.](image)

Notice that $\phi$ induces a one-to-one mapping of strata of $G_1$ to strata of $G_2$ such that $G \mapsto R$, $R \mapsto B$, and $B \mapsto G$. Furthermore, the number of red vertices in $G_1$ is equal to the number of blue vertices in $G_2$, the number of blue vertices
in \( G_1 \) is the same as the number of green vertices in \( G_2 \), and the number of green vertices in \( G_1 \) is equal to the number of red vertices in \( G_2 \). On the other hand, consider the one-to-one mapping \( \theta : V(G_1) \rightarrow V(G_2) \) defined by

\[
\theta : \begin{align*}
v_1 &\mapsto u_3 \\
v_2 &\mapsto u_4 \\
v_3 &\mapsto u_2 \\
v_4 &\mapsto u_5 \\
v_5 &\mapsto u_1.
\end{align*}
\]

Clearly \( \theta \) preserves adjacency; however, \( \theta \) does not preserve stratification. The vertices \( v_2 \) and \( v_4 \), which belong to the same color class in \( G_1 \), are mapped to \( u_4 \) and \( u_5 \), respectively, which belong to distinct color classes in \( G_2 \).

Figure 4 illustrates all stratified graphs on four vertices. The number of distinct 1-, 2-, 3-, and 4-stratified graphs are 11, 39, 24, and 11, respectively.

**Theorem 1** The number \( D_u \) of unicolored edges and the number \( D_b \) of bicolored edges are invariant under stratified graph isomorphism.

**Proof:** Let \( G_1 \) and \( G_2 \) denote two isomorphic \( k \)-stratified graphs, and let \( \phi \) denote an isomorphism from \( G_1 \) to \( G_2 \). The number \( D_u(G_1) \) is the number of edges \( xy \) in \( G_1 \) such that \( \text{color}(x) = \text{color}(y) \). Since \( \phi \) preserves both adjacency and stratification, \( \phi(x)\phi(y) \in E(G_2) \) and \( \text{color}(\phi(x)) = \text{color}(\phi(y)) \). Therefore, \( D_u(G_1) = D_u(G_2) \). But, \( D_b(G_1) = m - D_u(G_1) = m - D_u(G_2) = D_b(G_2) \).

An *automorphism* of a stratified graph \( G \) is an isomorphism of \( G \) with itself. In this case, the isomorphism \( \phi \) induces a permutation of the strata of \( G \). We thus have the following:

**Theorem 2** Let \( G \) be a stratified graph. The set \( A(G) \) of all automorphisms of \( G \) forms a group (under the operation of composition).
Figure 4. There are 89 Stratified Graphs on 4 Vertices.
Proof: Let $e$ denote the identity permutation of the vertices of $G$. Then clearly $e \in A(G)$. Since composition is associative, for $\alpha, \beta, \gamma \in A(G)$, it follows that $(\alpha\beta)\gamma = \alpha(\beta\gamma)$. For each $\alpha \in A(G)$, let $\alpha^{-1}$ denote the inverse permutation of the vertex set of $G$. We show that $\alpha^{-1} \in A(G)$. That is, for $u, v \in V(G)$, we have that $uv \in E(G)$ if and only if $\alpha^{-1}(u)\alpha^{-1}(v) \in A(G)$, and $\text{color}(u) = \text{color}(v)$ if and only if $\text{color}(\alpha^{-1}(u)) = \text{color}(\alpha^{-1}(v))$. Let $x = \alpha^{-1}(u)$ and $y = \alpha^{-1}(v)$. Since $\alpha \in A(G)$, it follows that $xy \in E(G)$ if and only if $\alpha(x)\alpha(y) \in E(G)$. Thus $\alpha^{-1}(u)\alpha^{-1}(v) \in E(G)$ if and only if $\alpha(\alpha^{-1}(u))\alpha(\alpha^{-1}(v)) \in E(G)$. This implies that $\alpha^{-1}(u)\alpha^{-1}(v) \in E(G)$ if and only if $uv \in E(G)$. Analogously, we can show that $\text{color}(u) = \text{color}(v)$ if and only if $\text{color}(\alpha^{-1}(u)) = \text{color}(\alpha^{-1}(v))$. This completes the proof that $A(G)$ is a group. □

Since every automorphism of $G$ that induces a permutation of the strata of $G$ is also an automorphism of the underlying graph $G'$, the automorphism group of $G$ is a subgroup of the automorphism group of $G'$.

**Theorem 3** The automorphism group $A(G)$ of a stratified graph $G$ is a subgroup of the automorphism group $A(G')$ of its underlying graph $G'$.

**Proof:** By Theorem 2, $A(G)$ is a group. Hence, it suffices to show that every automorphism $\alpha$ of $A(G)$ is also an automorphism of $A(G')$. Since $\alpha$ belongs to $A(G)$, it follows that $\alpha$ preserves adjacency and nonadjacency. Therefore, $\alpha \in A(G')$. □

This suggests a question. Given a finite group $\Gamma'$ and a subgroup $\Gamma$ of $\Gamma'$, does there exist a stratified graph $G$ such that $A(G) \cong \Gamma$ and $A(G') \cong \Gamma'$?

**Theorem 4** Let $G$ and $H$ be two isomorphic $k$-stratified graphs, that is, let $G \cong_k H$. Then $A(G) \cong A(H)$ under conjugation.
Proof: Since $G \cong_k H$, there exists a bijective function $\phi : V(G) \to V(H)$. Consider the mapping $I : A(G) \to A(H)$ given by $I(\alpha) = \phi \alpha \phi^{-1}$ for all $\alpha \in A(G)$. Then

\[
I(\alpha \beta) = \phi(\alpha \beta)\phi^{-1} = (\phi \alpha \phi^{-1})(\phi \beta \phi^{-1}) = I(\alpha)I(\beta)
\]

Hence, $I$ is a homomorphism. Suppose that $I(\alpha) = I(\beta)$. Then

\[
\phi \alpha \phi^{-1} = \phi \beta \phi^{-1} \quad \phi^{-1}(\phi \alpha \phi^{-1}) \phi = \phi^{-1}(\phi \beta \phi^{-1}) \phi \quad \alpha = \beta,
\]

which implies that $I$ is one-to-one.

For each $\gamma \in A(G)$, note that $I(\phi^{-1} \gamma \phi) = \phi(\phi^{-1} \gamma \phi) \phi = \gamma$. That is, $I$ is onto. Therefore, $I$ is an isomorphism of $A(G) \to A(H)$. □

Let $G$ be a $k$-stratified graph with the stratification $V(G) = S_1 \cup S_2 \cup \ldots \cup S_k$, and let $H$ denote a $k$-stratified graph with $V(H) = V(G)$ having the same stratification as that of $G$, that is, $V(H) = S_1 \cup S_2 \cup \ldots \cup S_k$. Let $E(H) = E(G')$, that is, $uv \in E(H)$ if and only if $uv \not\in E(G)$. We define a $k$-stratified graph $\bar{G}$ as a complement of the $k$-stratified graph $G$ if and only if $\bar{G} \cong_k H$. Consider the 2-stratified graphs drawn in the Figure 5.

Clearly, $H$ is a complement of $G$. However, since $G \cong_k H$, it follows that $G$ is a self-complementary 2-stratified graph.
Figure 5. A Self-complementary 2-stratified Graph.

**Theorem 5** For any $k$-stratified graph $G$ and its complement $\bar{G}$, $A(G) \cong A(\bar{G})$.

**Proof:** Let $H$ denote the $k$-stratified graph obtained by interchanging adjacency and nonadjacency in $G$. Clearly, $H$ is a complement of $G$. Next we show that if $\alpha \in A(G)$, then $\alpha \in A(H)$. If $\alpha \in A(G)$, then $\alpha$ is a permutation of $V(G)$ that preserves adjacency and stratification. But $\alpha$ preserves adjacency if and only if it preserves nonadjacency. Since $H$ has the same stratification as $G$, it follows that a permutation on $V(G)$ is an automorphism of $G$ if and only if it is an automorphism of $H$, implying that $A(G) \cong A(H)$. But since $\bar{G} \cong_k H$, Theorem 4 implies that $A(G) \cong A(\bar{G})$. □
CHAPTER II

DEGREE IN STRATIFIED GRAPHS

2.1 Introduction

In this section we investigate in more detail the concept of degree as it pertains to stratified graphs.

Recall that a sequence $\sigma : d_1, d_2, \ldots, d_n$ of nonnegative integers is called a degree sequence of a graph $G$ if the vertices of $G$ can be labeled $v_1, v_2, \ldots, v_n$ so that $\text{deg}(v_i) = d_i$ for all $i$. A finite sequence of nonnegative integers is graphical if it is the degree sequence of some graph. Havel [V.55] and Hakimi [L.62] described a necessary and sufficient condition for a finite sequence of nonnegative integers to be graphical.

In the next section we define the concepts of color pattern and arrangement and characterize those arrangements that are graphical. These notions play a similar role for stratified graphs to those of degree and degree sequences for graphs. In the following section we discuss the existence of stratified graphs with fixed color patterns for each color. This is similar to the concept of regular graphs. So we refer to them as color-regular graphs.

In the final section of this chapter we show that it is unlikely to obtain a simple characterization of graphical degree-vector multisets. However, in several special cases we find such characterizations.
2.2 Arrangements

A color pattern \( \mathcal{P} \) is a vector of the type \((X; d_1, d_2, \ldots, d_k)\), where \(X\) is a color (an element of some finite predetermined set) and each \(d_i\) \((1 \leq i \leq k)\) is a nonnegative integer. The color \(X\) is referred to as the 0th coordinate of \(\mathcal{P}\) and \(d_i\) \((1 \leq i \leq k)\) is the \(i\)th coordinate of \(\mathcal{P}\). In a stratified graph \(G\), every vertex \(v\) of \(G\) has a color pattern, namely \(\mathcal{P}(v) = (\text{color}(v); d_1, d_2, \ldots, d_k)\), where \(v\) is adjacent to \(d_i\) vertices of \(S_i\) \((1 \leq i \leq k)\).

A multiset \(\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n\}\) of color patterns is referred to as an arrangement. An arrangement is graphical if it is the multiset of color patterns of the vertices of some stratified graph.

A graphical arrangement need not determine a unique stratified graph. For example, \(\{(R;1,0),(R;1,1),(B;2,0),(R;0,1),(R;0,1),(B;1,0)\}\) is the arrangement of the nonisomorphic stratified graphs shown in Figure 6.

![Figure 6. Two Nonisomorphic Stratified Graphs With the Same Arrangement.](image)

We are now concerned with the problem of determining those arrangements that are graphical. Some necessary conditions are now presented.

**Proposition 1** Let \(A = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n\}\) be a graphical arrangement. For each stratified graph \(G\) having arrangement \(A\) and each color \(X\) of \(G\), the number of color patterns \(\mathcal{P}_i\) in \(A\) having 0th coordinate \(X\) is the number of \(X\)-colored vertices
Proposition 2 Let $A$ be a graphical arrangement. The sequence of nonnegative integers obtained by summing the color degrees $d_j$ in each color pattern $P_i$ is a graphical sequence.

Proof: Let $G$ be a stratified graph with arrangement $A$. Then the underlying graph of $G$ possesses the sequence so produced as its degree sequence.

Given a stratified graph $G$ with vertices $v_1, v_2, \ldots, v_n$ and arrangement $A = \{P_1, P_2, \ldots, P_n\}$, let $\sigma_{X_j}$ denote the sequence of nonnegative integers in the $j$th coordinate of the color patterns $P_i$ for all vertices $v_i$ with $\text{color}(v_i) = X_j$. Hence, $\sigma_{X_j}$ is a degree sequence of the $X_j$-colored subgraph of $G$.

Lemma 1 Let $A$ be a graphical arrangement. For each positive integer $j$ ($1 \leq j \leq k$), the sequence $\sigma_{X_j}$ is graphical.

For example, consider the arrangement $A = \{(X_1; 1,1), (X_1; 2,1), (X_1; 1,0), (X_2; 1,1), (X_2; 1,1)\}$. The arrangement $A$ can also be described as $A = \{(R; 1,1), (R; 2,1), (R; 1,0), (B; 1,1), (B; 1,1)\}$, where $X_1$ is red and $X_2$ is blue. Suppose that there is a 2-stratified graph $G$ with such an arrangement. Then, from the information given in the arrangement $A$, we can find the degree sequence for the red and the blue subgraphs of $G$. The degree sequence $\sigma_R$ for the red subgraph consists of the entries in the first coordinates of the arrangements for the red vertices. Thus, $\sigma_R : 1, 2, 1$ for the example given above. Similarly, the degree sequence $\sigma_B$ for the blue subgraph consists of the entries in the second coordinates of the arrangements for the blue vertices. Thus, $\sigma_B : 1, 1$ for our example.

To construct $G$ corresponding to $A$, we first construct a graph $G_R$ with degree sequence $\sigma_R$ (using the Havel-Hakimi theorem [MGL81], say) and color
the vertices of $G_R$ red. Then we construct a graph $G_B$ with degree sequence $\sigma_B$ and color all its vertices blue. Finally, we join some red vertices to some blue vertices such that the red degree of the blue vertices and the blue degree of the red vertices are as specified in their color patterns. Clearly, this means that the sum of the red degrees of the blue vertices is the sum of the blue degrees of the red vertices. Furthermore, there must exist at most one edge between a pair of red and blue vertices. This implies that the maximum red degree of a blue vertex must not exceed the number of red vertices and vice-versa (i.e., it must satisfy the condition in Lemma 3). A graph realizing the above arrangement is presented in Figure 7.

![Figure 7. A 2-stratified Graph Corresponding to the Arrangement A.](image)

**Proposition 3** Let $G$ be a $k$-stratified graph, $k \geq 1$. For each $v_i \in V(G)$, it follows that $d_j^i \leq |S_j|$, where $1 \leq j \leq k$. That is, no vertex is adjacent to more than $|S_j|$ vertices in stratum $S_j$.

Let $\sigma_{X_i,X_j}$ denote the sequence $a_1, a_2, \ldots, a_r$ ($r \geq 1$), of nonnegative integers where $|S_i| = r$ and the numbers $a_m$ for $1 \leq m \leq r$ are the $X_j$-degrees of the $X_i$-colored vertices (i.e., the integers in the $j$th coordinates of the color patterns $P_h$, where $\text{color}(v_h) = X_i$). Note that $m$ and $h$ may be different since the $X_j$-degree of some $X_i$-colored vertex may be 0. If the $X_j$-degree of some
X_i-colored vertex is 0, then there is no element in the sequence $\sigma_{X_i}x_j$ corresponding to the $X_j$-degree of this vertex. We can similarly construct the sequence $\sigma_{X_j}x_i : b_1, b_2, \ldots, b_t$ of positive integers. Lemma 3 implies the following necessary conditions: $a_h \leq t$, for $1 \leq h \leq r$ and $b_h \leq r$, for $1 \leq h \leq t$. Furthermore, it is necessary that $\Sigma_i a_i = \Sigma_i b_i$.

We now state a necessary and sufficient condition for a pair $s_1, s_2$ of finite sequences of nonnegative integers to be bigraphical, that is, a necessary and sufficient condition for the existence of a bipartite graph $G$ with partite sets $V_1$ and $V_2$ such that $s_1$ and $s_2$ are the degrees in $G$ of vertices in $V_1$ and $V_2$, respectively.

**Lemma 2** The sequences $s_1 : a_1, a_2, \ldots, a_r$ and $s_2 : b_1, b_2, \ldots, b_t$ of nonnegative integers with $r \geq 2$, $a_1 \geq a_2 \geq \ldots \geq a_r$, $b_1 \geq b_2 \geq \ldots \geq b_t$, $0 < a_1 \leq t$, and $0 < b_1 \leq r$ are bigraphical if and only if the sequences $s'_1 : a_2, a_3, \ldots, a_r$ and $s'_2 : b_1 - 1, b_2 - 1, \ldots, b_{a_1} - 1, b_{a_1+1}, \ldots, b_t$ are bigraphical.

**Proof:** First assume that $s_1$ and $s_2$ as stated above are bigraphical. Then there exists a bipartite graph with $s_1$ and $s_2$ as the degree sequences for its two partite sets. Let $G$ be one of these graphs. Suppose that $V(G)$ is partitioned into partite sets $V_1$ and $V_2$, where $V_1' = \{u_1, u_2, \ldots, u_r\}$, $V_2' = \{v_1, v_2, \ldots, v_t\}$, $\deg(u_i) = a_i$ for $1 \leq i \leq r$, $\deg(v_j) = b_j$ for $1 \leq j \leq t$, and the sum of the degrees of the vertices adjacent with $u_1$ is maximum. We claim that $u_1$ is adjacent to vertices having degrees $b_1, b_2, \ldots, b_{a_1}$. Suppose this is not the case. Then there exist vertices $v_x, v_y \in V_2$ with $b_x > b_y$ with $u_1$ adjacent to $v_y$ but not to $v_x$. Furthermore, there exists $u_z \in V_1$ such that $u_zv_x \in E(G)$ and $u_zv_y \not\in E(G)$. Define $G'$ to be the graph $G - u_zv_x - u_1v_y + u_zv_y + u_1v_x$. Observe that $G'$ is bipartite, $V_1$ has degree sequence $s_1$, and $V_2$ has degree sequence $s_2$. Note that the sum of the degrees of the vertices adjacent to $u_1$ is larger in $G'$ than in $G$, producing a contradiction.
Thus $u_1$ is adjacent to vertices with degrees $b_1, b_2, \ldots, b_{a_1}$. Hence $G - u_1$ is a bipartite graph with partite sets $V'_1 = \{u_2, u_3, \ldots, u_r\}$ and $V'_2 = \{v_1, v_2, \ldots, v_t\}$. Also, the degree sequence of $V'_1$ is $s'_1: a_2, a_3, \ldots, a_r$ while the degree sequence of $V'_2$ is $s'_2: b_1 - 1, b_2 - 1, \ldots, b_{a_1 - 1}, b_{a_1 + 1}, \ldots, b_t$. Therefore, $s'_1$ and $s'_2$ are bigraphical.

For the converse, assume the sequences $s'_1: a_2, a_3, \ldots, a_r$ and $s'_2: b_1 - 1, b_2 - 1, \ldots, b_{a_1 - 1}, b_{a_1 + 1}, \ldots, b_t$ are bigraphical. Then there exists a bipartite graph $G$ having partite sets $V_1$ and $V_2$ where $V_1 = \{u_2, u_3, \ldots, u_r\}$, $\deg(u_i) = a_i$ for $2 \leq i \leq r$, $V_2 = \{v_1, v_2, \ldots, v_t\}$, $\deg(v_j) = b_j - 1$ for $1 \leq j \leq a_1$, and $\deg(v_j) = b_j$ for $a_1 + 1 \leq j \leq t$. Add a vertex $u_1$ to $V_1$ and join $u_1$ to the vertices $v_1, v_2, \ldots, v_{a_1}$. The resulting graph is bipartite with partite sets $V'_1 = \{u_1, u_2, \ldots, u_r\}$ and $V'_2 = \{v_1, v_2, \ldots, v_t\}$, and degree sequences $s_1: a_1, a_2, \ldots, a_r$, $s_2: b_1, b_2, \ldots, b_t$, respectively. That is, $s_1$ and $s_2$ are bigraphical.

We are now ready to state our main result of this section

**Theorem 6** An arrangement $\mathcal{A} = \{P_1, P_2, \ldots, P_n\}$ is graphical if and only if for each $i$ with $1 \leq i \leq k$, the sequences $\sigma_{X_i}$ are graphical and for each pair $i, j$ of integers with $1 \leq i \neq j \leq k$ the pairs $\sigma_{X_i, X_j}, \sigma_{X_j, X_i}$ of sequences are bigraphical.

**Proof:** Certainly the stated conditions are necessary for the arrangement $\mathcal{A}$ to be graphical. We now verify the sufficiency. Assume then that the sequences $\sigma_{X_i}$ ($1 \leq i \leq k$) are graphical and that the pairs $\sigma_{X_i, X_j}, \sigma_{X_j, X_i}$ ($1 \leq i \neq j \leq k$) of sequences are bigraphical. We now construct a $k$-stratified graph $G$ with arrangement $\mathcal{A}$.

For $1 \leq i \leq k$, assume that there are $n_i$ color patterns with first coordinate $X_i$. So $\sum n_i = n$. Define $N_0 = 0$, $N_t = \sum_{i=1}^{t} n_i$, $1 \leq t \leq k$, and $V(G) = \{v_1, v_2, \ldots, v_n\}$ such that for each $j$ with $1 \leq j \leq n$, $\text{color}(v_j) = X_i$ if $t$ is the smallest positive integer such that $j \leq N_t$, and if $\text{color}(v_1) = \text{color}(v_j) = X_i$.
then in \(<S_i>\), \(\deg(v_i) \leq \deg(v_j)\) implies that \(i \geq j\). Consequently, the vertices \(v_1, v_2, \ldots, v_n\) are colored \(X_1\) where the vertices are so labeled that their degrees in \(<S_1>\) form a nonincreasing sequence. In general, for \(1 \leq i \leq k\), the vertices \(v_{N_i-1+1}, v_{N_i-1+2}, \ldots, v_{N_i}\) are colored \(X_1\) where the vertices are so labeled that their degrees in \(<S_i>\) form a nonincreasing sequence.

We now consider distinct integers \(i\) and \(j\) with \(1 \leq i, j \leq k\) and the pair \(\sigma_{x_i,x_j}, \sigma_{x_j,x_i}\) of bigraphical sequences. Hence there exists a bipartite graph \(G_{i,j}\) with partite sets \(S'_i\) and \(S'_j\) with \(|S'_i| = n_i\) and \(|S'_j| = n_j\). After possible permutation of the vertices of \(S'_i\) and \(S'_j\), we have \(S'_i = \{v'_{N_i-1+1}, v'_{N_i-1+2}, \ldots, v'_{N_i}\}\), where in \(G_{i,j}\) we have that \(\deg(v'_{N_i-1+t})\) is the \(j\)th term in the color pattern of \(P(v'_{N_i-1+t})\) for \(t = 1, 2, \ldots, n_i\) and where \(\deg(v'_{N_j-1+r})\) is the \(i\)th term in the color pattern of \(P(v'_{N_j-1+r})\) for \(r = 1, 2, \ldots, n_j\). We then superimpose \(G_{i,j}\) so that \(S'_i\) and \(S_i\) are identified, as are \(S'_j\) and \(S_j\), with \(v'_{N_{i-1}+t} \rightarrow v_{N_{i-1}+t}\) and \(v'_{N_{j-1}+r} \rightarrow v_{N_{j-1}+r}\) for \(i \leq t \leq n_i\) and \(1 \leq r \leq n_j\). When this is done for all pairs \(i, j\), the desired \(k\)-stratified graph is produced. □

We now illustrate the proof of the previous theorem for the arrangement \(A = \{(R; 2, 2), (R; 1, 4), (B; 1, 2), (R; 3, 2), (R; 1, 2), (B; 3, 1), (B; 3, 3), (B; 4, 2), (R; 3, 1)\}\). We now order the terms of \(A\) so that those color patterns with color red are listed first and those with color blue are listed second. Furthermore, the color patterns with color red are listed so that the sequence of 1st coordinates are nonincreasing and the color patterns with color blue are listed so that the sequence of 2nd coordinates are nonincreasing. We now have the following:

\[
P(v_1) = (R; 3, 2) \quad P(v_2) = (R; 3, 1) \quad P(v_3) = (R; 2, 2) \\
P(v_4) = (R; 1, 2) \quad P(v_5) = (R; 1, 4) \quad P(v_6) = (B; 3, 3) \\
P(v_7) = (B; 4, 2) \quad P(v_8) = (B; 1, 2) \quad P(v_9) = (B; 3, 1)
\]

Hence we have the sequences \(\sigma_R : 3, 3, 2, 1, 1\) and \(\sigma_B : 3, 2, 2, 1\), both of
which are graphical. The graphs $G_R$ and $G_B$ of Figure 8 have degree sequences $\sigma_R$ and $\sigma_B$, respectively. We now consider the sequences $\sigma_{R,B} : 2, 1, 2, 2, 4$ and $\sigma_{B,R} : 3, 4, 1, 3$. An application of Lemma 5 shows that the pair $\sigma_{R,B}, \sigma_{B,R}$ of sequences are bigraphical. A bipartite graph with these bigraphical sequences is shown in Figure 9.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
By identifying $v_i$ and $v'_i$ ($1 \leq i \leq 9$) in the two graphs of Figure 9 and the bipartite graph of Figure 8, we produce a 2-stratified graph $G$ with arrangement $\mathcal{A}$, shown in Figure 10.

![Graph G](image)

Figure 10. A 2-stratified Graph With Arrangement $\mathcal{A}$.

### 2.3 Color-Regular Stratified Graphs

A stratified graph is **color-regular** if every two vertices of the same color have the same color pattern. The 2-stratified graph $G$ of Figure 11 is color-regular; all red vertices have color pattern $(R; 3, 3)$ while every blue vertex has color pattern $(B; 2, 3)$.

The preceding example shows, therefore, that there exists a color-regular 2-stratified graph with color patterns $(R; 3, 3)$ and $(B; 2, 3)$. In this section we determine for all patterns $(R; r_1, b_1)$ and $(B; r_2, b_2)$ for which there exists a color-regular 2-stratified graph having these color patterns. In the case where such a 2-stratified graph exists, we also determine the minimum order of such a 2-stratified graph. We also investigate this problem for color-regular $k$-stratified graphs, where $k \geq 3$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Theorem 7 Let $r_1, r_2, b_1, b_2$ be nonnegative integers. There exists a color-regular 2-stratified graph whose vertices have color patterns $(R; r_1, b_1)$ and $(B; r_2, b_2)$ unless exactly one of $b_1$ and $r_2$ is 0. Furthermore, the smallest order of a color-regular 2-stratified graph with color patterns $(R; r_1, b_1)$ and $(B; r_2, b_2)$ is $x + y$, where $x$ and $y$ are the smallest positive integers satisfying the following:

(I) $x \geq \max\{r_2, r_1 + 1\}$, $y \geq \max\{b_1, b_2 + 1\}$

(II) If $r_1$ is odd, then $x$ is even; and if $b_2$ is odd, then $y$ is even.

(III) $xb_1$ and $yr_2$ are equal multiples of $\text{lcm}(b_1, r_2)$.

Proof: First, suppose that exactly one of $b_1$ and $r_2$ is 0, say $b_1 > 0$ and $r_2 = 0$. This says that every red vertex is adjacent to $b_1$ blue vertices but no blue vertex is adjacent to a red vertex, which is impossible. If $b_1 = r_2 = 0$, the result is trivial. So we assume that $b_1 > 0$ and $r_2 > 0$. Assume that such a 2-stratified graph exists, and let $S_1 = \{u_1, u_2, \ldots, u_x\}$ and $S_2 = \{v_1, v_2, \ldots, v_y\}$ denote the strata of a color-regular 2-stratified graph $G$ with color patterns $(R; r_1, b_1)$ and $(B; r_2, b_2)$. Since all blue vertices are adjacent to $r_2$ red vertices, $x \geq r_2$; and since all red vertices are adjacent to $r_1$ red vertices, $x \geq r_1 + 1$. Hence, $x \geq$: 

![Diagram of a color-regular 2-stratified graph](image-url)
\[
\max\{r_2, r_1 + 1\}. \text{ Similarly, } y \geq \max\{b_1, b_2 + 1\}. \text{ This establishes the necessity of (I).}
\]

The red subgraph of \(G\) is an \(r_1\)-regular and the blue subgraph of \(G\) is a \(b_2\)-regular graph. Since the number of vertices of odd degree in any graph is always even, (II) holds.

The number of edges between \(S_1\) and \(S_2\) is \(b_1x = r_2y\). So this number is a multiple of \(m = \text{lcm}(b_1, r_2)\). If \(b_1x' = r_2y' = m\), then \(b_1x = km\) for some positive integer \(k\). But \(r_2y = b_1x\), so \(r_2y = km\) as well, establishing the necessity of (III).

For the converse, we show that for the smallest positive integers \(x\) and \(y\) satisfying the given three conditions, there is a color-regular 2-stratified graph with \(x\) red vertices and \(y\) blue vertices with color patterns \((R, r_1, b_1)\) and \((B, r_2, b_2)\). Let \(S_1 = \{u_1, u_2, \ldots, u_x\}\) denote the red vertices and \(S_2 = \{v_1, v_2, \ldots, v_y\}\) denote the blue vertices. Since \(x \geq r_1 + 1\) and \(x\) meets the parity requirement (i.e., \(x\) is even if \(r_1\) is odd), we can construct an \(r_1\)-regular graph of order \(x\) (see [CO93], for example). Similarly, we show that it is possible to construct a \(b_2\)-regular graph of order \(y\). After constructing an \(r_1\)-regular graph on the \(x\) vertices of \(S_1\) and a \(b_2\)-regular graph on the \(y\) vertices of \(S_2\), we then join \(u_1\) to \(v_1, v_2, \ldots, v_y\). Since \(y \geq b_1\), no multiple edges are created. Then we join \(u_2\) to \(v_{b_1+1}, v_{b_1+2}, \ldots, v_{2b}\), where the subscripts \(b_1 + 1, b_1 + 2, \ldots, 2b\) are expressed as one of the numbers \(1, 2, \ldots, y\) modulo \(y\). This procedure is continued with \(u_3, u_4, \ldots, u_x\). Since \(b_1x\) edges are equally distributed among the \(y\) vertices of \(S_2\) and \(y\) divides \(b_1x\), every vertex in \(S_2\) has the same degree. Since \(b_1x = r_2y\), this degree is \(r_2\) \(\Box\)

We now illustrate the previous theorem with an application. We use this result to determine the smallest 2-stratified color-regular graph in which every red vertex is adjacent with 9 red and 6 blue vertices, and each blue vertex is adjacent with 4 red and 8 blue vertices. In other words, we wish to find the minimum order
Figure 12. Construction of a Color-regular Graph With Color Patterns (R; 9, 6) and (B; 4, 8).

of a 2-stratified color-regular graph with color patterns (R; 9, 6) and (B; 4, 8). Thus, $r_1 = 9$, $b_1 = 6$, $r_2 = 4$, and $b_2 = 8$. Hence, $x \geq \max\{r_2, r_1 + 1\} = \max\{4, 9 + 1\} = 10$ and $y \geq \max\{b_1, b_2 + 1\} = \max\{6, 8 + 1\} = 9$. Since $\text{lcm}(b_1, r_2) = \text{lcm}(4, 6) = 12$, it follows that $6x = 4y = 12k$ for some positive integer $k$. That is, $x = 2k$ and $y = 3k$. Also, since $r_1$ is odd, $x$ is even. The smallest positive integer $k$ that satisfies all these conditions is $k = 5$. Therefore, the minimum order of a 2-stratified color-regular graph with color patterns (R; 9, 6) and (B; 4, 8) is 25. This graph $G$ has 10 red and 15 blue vertices. The red subgraph of this graph is isomorphic to $K_{10}$, and the blue subgraph of $G$ is an 8-regular graph of order 15. See Figure 12 for an illustration of the red-blue edges of $G$ following the construction outlined in the Theorem 7.

Now we consider the problem of determining the existence of color-regular $k$-stratified graphs with $k \geq 3$ having prescribed color patterns. Note that, in general, a 3-stratified color-regular graph will have a color pattern set of the type \{(R; r_1, b_1, g_1), (B; r_2, b_2, g_2), (G; r_3, b_3, g_3)\}. Let $x, y,$ and $z$ denote the number of red, blue, and green vertices, respectively. Since the number of edges joining a red vertex and a blue vertex can be expressed by $xb_1$ or $yr_2$, it follows that $xb_1 = yr_2$. In general, $x, y,$ and $z$ satisfy
The color pattern set \{((R; 2, 2, 2), (B; 3, 2, 3), (G; 5, 4, 2))\} leads to the inconsistent system of equations:

\[
\begin{align*}
2x &= 3y \\
2x &= 5z \\
3y &= 4z
\end{align*}
\]

Therefore, there cannot exist a 3-stratified color-regular graph with such a color pattern set.

However, the color pattern set \{((R; 2, 2, 2), (B; 3, 2, 3), (G; 5, 5, 2))\} leads to the system of equations:

\[
\begin{align*}
2x &= 3y \\
2x &= 5z \\
3y &= 5z
\end{align*}
\]

which is consistent. An application of the Theorem 7 to each pair of equations from this system yields the values \(x = 30, y = 20,\) and \(z = 12\). Hence, the smallest 3-stratified color-regular graph with the given color pattern set has 30 red, 20 blue, and 12 green vertices. We illustrate the construction of this graph in Figure 13.

All edges corresponding to the red, blue, and green subgraphs are omitted. All three of these subgraphs are cycles of their respective orders. Also, only a few of the edges between the three strata are shown. However, all bicolored edges for the leftmost vertices in each stratum are drawn.

In general, the existence of a color-regular 3-stratified graph with pre-
scribed color pattern set \{ (R; r_1, b_1, g_1), (B; r_2, g_2), (G; r_3, b_3, g_3) \} depends entirely on whether the system (1) of equations is consistent. A similar statement holds for color-regular $k$-stratified graphs for $k \geq 4$.

### 2.4 Degree-Vector Multisets

In Chapter I we defined the degree-vector multisets of a stratified graph $G$ as the multiset consisting of the degree vectors of the vertices of $G$. It is straightforward to determine a degree-vector multiset for $G$. On the other hand, if a multiset $\sigma = \{D^1, D^2, \ldots, D^n\}$ of $k$-vectors of nonnegative integers is given, then under what conditions is $\sigma$ the degree-vector multiset of some $k$-stratified graph? If such a graph exists, then we refer to $\sigma$ as a graphical multiset.

We are not able to find an efficient algorithm to solve the problem described above. Instead, we turn our attention to classifying this problem according to its tractability. Observe that in a 2-stratified graph the number of edges leaving the first stratum is equal to the number of edges entering the second stratum and vice-versa. Clearly, a similar condition holds for $k$-stratified graphs. We formally define this necessary condition later in the section and call it the stratification property. We will then show that the general problem of finding whether a multiset
σ = \{D^1, D^2, \ldots, D^n\} of k-vectors of nonnegative integers has the stratification property is NP-complete. In this section we will also discuss several other related problems.

First, observe that the degree-vector multiset of a stratified graph does not provide sufficient information to determine a unique color density vector. For example, a 2-stratified graph with the degree-vector multiset \{(0,1), (0,1), (1,1), (1,1), (0,2)\} may have the color density vector (1,4) or (2,3), as illustrated in Figure 14.

\[
\begin{array}{cccccc}
B & B & R & B & B & R \\
0 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 2 & 1 & 1 & 1 \\
\end{array}
\begin{array}{cccccc}
R & B & B & B & R & B \\
0 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 2 & 1 & 1 & 1 \\
\end{array}
\]

Figure 14. Stratified Graphs With the Same Degree-vector Multiset but Having Distinct Density Vectors.

Now, we define the problem at hand in the notation and terminology of Garey and Johnson [MD79].

**GRAPHICABILITY**

**INSTANCE:** A multiset \(σ = \{D^1, D^2, \ldots, D^n\}\) of \(k\)-vectors of nonnegative integers such that \(D^i = (d^i_1, d^i_2, \ldots, d^i_k)\), where \(0 \leq d^i_j \leq n - 1, 1 \leq i \leq n, 1 \leq j \leq k\), and \(k \geq 2\).

**QUESTION:** Does there exist a \(k\)-stratified graph with degree-vector multiset \(σ\)?

The stratification problem is defined next.
STRATIFICATION

INSTANCE: A finite sequence $\sigma : D^1, D^2, \ldots, D^n$ of $k$-vectors of nonnegative integers such that $D^i = (d^i_1, d^i_2, \ldots, d^i_k)$, where $0 \leq d^i_j \leq n - 1$, $1 \leq i \leq n$, $1 \leq j \leq k$, and $k \geq 2$.

QUESTION: Is there a function $f : \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\}$ such that for each pair $(j_1, j_2)$ of integers with $1 \leq j_1 \neq j_2 \leq k$,

\[ \sum_{i \in f^{-1}(j_1)} d^i_{j_2} = \sum_{i \in f^{-1}(j_2)} d^i_{j_1}. \]  \hspace{1cm} (2)

The stratification problem is shown to be NP-complete by restriction. That is, we specify additional restrictions on the instances of the stratification problem so that the resulting restricted problem is equivalent to the partition problem which is well-known to be NP-complete [MD79]. The partition problem is described as follows.

PARTITION

INSTANCE: A finite set $A$ and a size $s(a) \in \mathbb{Z}^+$ for each $a \in A$.

QUESTION: Is there a subset $A' \subseteq A$ such that

\[ \sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a) ? \]

A subproblem of the stratification problem where $k = 2$ is stated below.

2-STRATIFICATION

INSTANCE: A multiset $\sigma = \{D^1, D^2, \ldots, D^n\}$ of $k$-vectors of nonnegative integers such that $D^i = (d^i_1, d^i_2)$, where $0 \leq d^i_1, d^i_2 \leq n - 1$, and $1 \leq i \leq n$.
QUESTION: Is there a function $f : \{1, 2, \ldots, n\} \rightarrow \{1, 2\}$ where $1 \leq i \leq n$ and $j = 1, 2$ such that

$$\sum_{i \in f^{-1}(1)} d_i^2 = \sum_{i \in f^{-1}(2)} d_i^1 ? \tag{3}$$

Theorem 8 The 2-STRATIFICATION problem is NP-complete.

Proof: It is easy to see that 2-STRATIFICATION $\in$ NP because a non-deterministic algorithm need only guess the function $f$ and check in polynomial time if the equation (3) holds.

We now restrict our attention only to those instances of the 2-stratification problem in which $\sum d_i^1 = \sum d_i^2$, $i = 1, 2, \ldots, n$. We transform PARTITION to this subproblem. Let the set $A$ and given size $s(a_i)$ for each $a_i \in A$ constitute an arbitrary instance of PARTITION, let $B = \sum_{a_i \in A} s(a_i)$ and $n = |A|$. The basic units of the PARTITION instance are the individual elements $a_i \in A$. The local replacement for each $a_i \in A$ is a single vector $D_i = (d_i^1, d_i^2)$ with $d_i^1 + d_i^2 = s(a_i)$. Let $t$ represent a variable counting the rank of $s(a_i)$ with odd value in the list of $a_i \in A$. That is, $t = 1$ for the $s(a_i)$ that is odd and has the smallest value for $i$. And, $t = 2$ for the next $s(a_i)$ that is odd, and so on. Let $d_i^1 = \lfloor s(a_i)/2 \rfloor$, $d_i^2 = \lfloor s(a_i)/2 \rfloor$ if $t$ is odd, and $d_i^1 = \lfloor s(a_i)/2 \rfloor$, $d_i^2 = \lfloor s(a_i)/2 \rfloor$ if $t$ is even. Note that if $B$ is odd, then the answer to the PARTITION problem is automatically negative. Thus, we are interested only in those instances of the PARTITION problem in which $B$ is even. Clearly, the PARTITION problem remains NP-complete for this subset of instances as well. But, since $B$ is even, the number of $a_i$ with odd $s(a_i)$ in $A$ is also even. Therefore, our scheme of assigning values to $d_i^j$ results in $\sum_{i=1}^n d_i^1 = \sum_{i=1}^n d_i^2 = B/2$. Clearly, this instance can be constructed in polynomial time from the PARTITION instance.
Suppose that there is a function $f$ such that

$$\sum_{i \in f^{-1}(1)} d_i^2 = \sum_{i \in f^{-1}(2)} d_i^1.$$  

Since $f$ is a function, $f^{-1}(1) \cup f^{-1}(2) = \{1, 2, \ldots, n\}$. From this transformation, we have $\sum_{i=1}^{n} d_i^1 = \sum_{i=1}^{n} d_i^2 = B/2$. Thus,

$$\sum_{i \in f^{-1}(1)} d_i^1 + \sum_{i \in f^{-1}(2)} d_i^1 = \sum_{i \in f^{-1}(1)} d_i^2 + \sum_{i \in f^{-1}(2)} d_i^2.$$  

Hence,

$$\sum_{i \in f^{-1}(1)} d_i^1 + \sum_{i \in f^{-1}(2)} d_i^2 = \sum_{i \in f^{-1}(1)} d_i^1 + \sum_{i \in f^{-1}(2)} d_i^2.$$  

That is,

$$\sum_{i \in f^{-1}(1)} (d_i^1 + d_i^2) = \sum_{i \in f^{-1}(2)} (d_i^1 + d_i^2)$$

which implies that

$$\sum_{i \in f^{-1}(1)} s(a_i) = \sum_{i \in f^{-1}(2)} s(a_i).$$

Let $A' = \{a_i | i \in f^{-1}(1)\}$ and $A - A' = \{a_i | i \in f^{-1}(2)\}$. Then

$$\sum_{a_i \in A'} s(a_i) = \sum_{a_i \in A - A'} s(a_i).$$

Therefore, if there is a positive solution to the 2-STRATIFICATION problem, there is also a positive solution to the corresponding PARTITION problem.

Conversely, suppose that there is a positive solution to the PARTITION problem, that is, suppose there is a subset $A' \subset A$ such that

$$\sum_{a_i \in A'} s(a_i) = \sum_{a_i \in A - A'} s(a_i).$$
Define $f(i) = 1$ for all $i$ such that $a_i \in A'$ and $f(i) = 2$, otherwise. Hence,

$$\sum_{i \in f^{-1}(1)} s(a_i) = \sum_{i \in f^{-1}(2)} s(a_i).$$

Thus,

$$\sum_{i \in f^{-1}(1)} (d_i^1 + d_i^2) = \sum_{i \in f^{-1}(2)} (d_i^1 + d_i^2),$$

which implies that

$$\sum_{i \in f^{-1}(1)} d_i^1 + \sum_{i \in f^{-1}(2)} d_i^2 = \sum_{i \in f^{-1}(2)} d_i^1 + \sum_{i \in f^{-1}(2)} d_i^2. \quad (4)$$

Hence again $\sum_{i=1}^{n} d_i^1 = \sum_{i=1}^{n} d_i^2 = B/2$, that is,

$$\sum_{i \in f^{-1}(1)} d_i^1 + \sum_{i \in f^{-1}(1)} d_i^1 = \sum_{i \in f^{-1}(2)} d_i^2 + \sum_{i \in f^{-1}(2)} d_i^2. \quad (5)$$

Thus, from (4) and (5) we have,

$$\sum_{i \in f^{-1}(1)} d_i^2 = \sum_{i \in f^{-1}(2)} d_i^1.$$

Hence, the function $f$ satisfies the requirements of the 2-STRATIFICATION problem.

Thus, the desired subset $A'$ exists for the instance of PARTITION if and only if there is a function $f$ satisfying the requirements of the 2-STRATIFICATION problem. \(\square\)

**Corollary 1** The STRATIFICATION problem is NP-complete.

**Proof:** It is easy to see that STRATIFICATION $\in$ NP because a nondeterministic algorithm need only guess the function $f$ and check in polynomial time...
if the $c(n, 2)$ equations (3) hold. The NP-completeness of this problem directly follows from that of the 2-STRATIFICATION problem. □

It is easy to see that the GRAPHICABILITY $\in$ NP because a nondeterministic algorithm need only guess a color for each $D^i$ and assign it to that $D^i$. The result is a multiset of color patterns. We can then use Theorem 6 from the preceding section to determine if this multiset of color patterns is graphical.

In light of this, we now turn our attention to recognizing the GRAPHICABILITY of a restricted subset of $k$-vectors. We characterize $k$-vector multisets that are the degree-vector multisets of a few of the well-known subclasses of 2-stratified graphs. Complete graphs form one of the most basic subclasses of graphs. We determine those 2-vector multisets that are degree-vector multisets of 2-stratified complete graphs.

**Theorem 9** Let $\sigma$ be a multiset of $n$ ordered pairs of nonnegative integers. Then $\sigma$ is the degree-vector multiset of a 2-stratified complete graph of order $n$ if and only if $n = r + b$ for nonnegative integers $r$ and $b$, and $\sigma$ consists of $r$ vectors of the type $(r - 1, b)$ and $b$ vectors of the type $(r, b - 1)$.

The 2-vector multisets that are degree-vector multisets of 2-stratified stars can also be easily characterized as follows.

**Theorem 10** Let $\sigma$ be a multiset of $n$ ordered pairs of nonnegative integers. Then $\sigma$ is the degree-vector multiset of a 2-stratified star of order $n$ if and only if $n = s + t + 1$ for nonnegative integers $s$ and $t$ and $\sigma$ consists of one vector of the type $(s, t)$, and $s + t$ vectors of the type $(0, 1)$ or $s + t$ vectors of the type $(1, 0)$.

Characterizing 2-vector multisets that are degree-vector multisets of 2-stratified cycles is more challenging. Before stating this result, we make a few
remarks and define additional terms. First, note that the color degree-vector of each vertex in a 2-stratified cycle is one of $(1, 1), (2, 0),$ and $(0, 2)$.

**Lemma 3** The sum of the $X$-color degrees of the vertices in a 2-stratified cycle equals twice the number of $X$-colored vertices in that cycle.

**Proof:** Each $X$-colored vertex contributes 1 to the $X$-color degree of exactly two other vertices namely its neighbors. Thus, in counting all the $X$-color degrees, we see that each $X$-colored vertex is accounted for exactly two times.

The previous lemma implies that for each color $X$, the $X$-color degree sum in a 2-stratified cycle is even. In a 2-stratified cycle $C$, we refer to a set of consecutive vertices of the same color as a *unicolor-block* of $C$. We use a sequence of characters R and B to describe the colors of the vertices in the cycle to give a *sequential listing of vertices* (*unicolor-blocks*). For the case of the listing of the unicolor-blocks, we assume that we always start at the beginning of the unicolor-block. The graph of Figure 15 has four unicolor-blocks.

![Figure 15. A Sequential Unicolor-block Listing: RRRBRBB.](image)

**Lemma 4** Every 2-stratified cycle consists of even number of unicolor-blocks.

**Proof:** If we list the unicolor-blocks sequentially, then each unicolor-block is immediately followed by one unicolor-block of the other color. In other words, the number of red color-blocks is equal to that of the blue color-blocks.

Observe that each unicolor-block of $C$ with two or more vertices gives rise to two terms equal to $(1, 1)$ in the degree-vector multisets of $C$. Also, note that if
the number of vertices in a unicolor-block is exactly 1, then this block contributes either (2, 0) or (0, 2) to the degree-vector multisets of $C$.

**Theorem 11** Let $\sigma$ be a multiset of $n$ ordered pairs of nonnegative integers. Then $\sigma$ is the degree-vector multiset of a 2-stratified cycle if and only if $\sigma$ consists of $m$ terms equal to $(1, 1)$, $a$ terms equal to $(2, 0)$, and $b$ terms equal to $(0, 2)$ where $m + a + b = n$ and one of the following conditions holds:

(I) $m = 0$ and $a = b \geq 2$,

(II) $m = 2$ and $a \neq b$,

(III) $m \equiv 0 \pmod{4}$ and $m \geq 4$,

(IV) $m \equiv 2 \pmod{4}$ and $m \geq 6$ and either $a \neq 0$ or $b \neq 0$.

**Proof:** If $\sigma$ is the degree-vector multiset of a 2-stratified cycle, then $m$ is even; otherwise we will have an odd number for each $X$-color degree sum which leads to a contradiction. Observe that if $m = 0$ then every unicolor-block of $C$ has exactly one vertex. Hence, $a = b$. If $m = 2$ then, there is exactly one unicolor-block with two or more vertices. Without loss of generality, assume this to be a red color-block. So, each blue color-block consists of only one vertex. Hence, there are exactly $b$ blue color-blocks in $C$. Since the number of blue color-blocks is equal to the number of red color-blocks, there are also $b$ red color-blocks in $C$. Because one of the red color-blocks has at least two vertices, if $a = b$, then only $a - 1$ of these are single vertex blocks. Observe that since all blue color-blocks consist of a single vertex, it follows that in this situation only single vertex red color-blocks give rise to $(0, 2)$ degree vectors. However, this implies that $a = b - 1$, which contradicts our assumption that $a = b$. Finally, since only unicolor-blocks
with two or more vertices give rise to terms equal to \((1, 1)\). If \(m \equiv 2 \,(\text{mod} \, 4)\), \(m \geq 6\) and \(a = b = 0\), then all unicolor-blocks must have exactly two vertices. But since \(m \equiv 2 \,(\text{mod} \, 4)\), this leads to an odd number of unicolor-blocks, which is impossible.

For the other direction, we describe how to construct a 2-stratified cycle with the prescribed degree-vector multiset for each case

Case (I): \(m = 0\) and \(a = b \geq 2\). Repeat the sequence RB exactly \(a\) times. See Figure 16 for an illustration of this case.

![Figure 16. Case (I), Alternating Colors: RBRB...RB.](image)

Case (II): \(m = 2\) and \(a \neq b\). Without loss of generality assume that \(a > b\). Start the construction of \(C\) with a red color-block having \(a - b\) vertices and follow this with \(b + 1\) alternating blue and red vertices. Notice that this results in \(a - b + 2(b + 1) = a + b + 2\) which is the correct number of vertices. Also, observe that the \(b + 1\) alternating blue and red vertices gives rise to exactly \(b\) terms equal to \((0, 2)\), \(b + 1\) terms of \((2, 0)\), and one \((1, 1)\). The unicolor-block of \(a - b\) red vertices contributes \(a - b - 1\) terms \((2, 0)\) and one \((1, 1)\). Therefore, the 2-stratified cycle so produced has the desired values for \(m\), \(a\), and \(b\). Figure 17 illustrates this construction for \(a = 5\) and \(b = 3\).

Case (III): \(m \equiv 0 \,(\text{mod} \, 4)\) and \(m \geq 4\). Let \(m = 4k\) for some positive integer \(k\). Then, repeat the sequence RRBB exactly \(k\) times. Now, insert exactly \(a\) red vertices in one red color-block and exactly \(b\) blue vertices in a blue color-block.
Case (IV): $m \equiv 2 \pmod{4}$, $m \geq 6$, and either $a \neq 0$ or $b \neq 0$. Then $m = 4k + 2$ for some positive integer $k$. Let $C$ be a 2-stratified cycle with $m$ terms $(1,1)$. This implies that there are at least $2k + 1$ unicolor-blocks in $C$. But since the number of unicolor-blocks in a 2-stratified cycle is always even, we must have at least $2k + 2$ unicolor-blocks in $C$. Since a unicolor-block with at least two vertices will give rise to two terms $(1,1)$ in the degree-vector multiset and since $m = 4k + 2$, at least one unicolor-block must contain at most one vertex. This implies that either $a \neq 0$ or $b \neq 0$.

Without loss of generality, assume that $a > 0$. To construct $C$, start with a blue vertex. Then list a sequence of $2k + 1$ unicolor-blocks, each of length 2, placed in a row with alternating colors such that the first unicolor-block is red. This will result in exactly $4k + 2$ terms $(1,1)$ and one $(2,0)$. Now, insert exactly $a - 1$ red vertices in a red unicolor-block and exactly $b$ blue vertices in a blue
Figure 19. Case 3, $a \neq 0$, or $b \neq 0$: BR...RB...B...RRBBRR.

unicolor-block. This will result in a 2-stratified cycle with $m$ terms $(1,1)$, $a$ terms $(2,0)$, and $b$ terms $(0,2)$. See Figure 19. □
CHAPTER III

DISTANCE IN STRATIFIED GRAPHS

3.1 Introduction

In many situations where graphs are used as mathematical models, distance between vertices is of primary importance. In [FF89] Buckley and Harary discuss a number of facility location problems, all of which involve distance in graphs. In determining where to locate an emergency facility such as a hospital or fire station, the response time between the facility and the location of a possible emergency is to be minimized. In deciding the position for a service facility such as a post office, power station, or employment office, the average travel time for the people in the district is to be minimized. When constructing a railroad line, pipeline, or superhighway, the distance from the new structure to each of the communities to be served is to be minimized. Each of these situations deals with the concept of centrality in various ways. However, the type of center involved differs for each of the three examples mentioned. Stratified graphs provide a natural means of modeling these situations. Centrality questions are now examined using stratified graphs and distance concepts.

In the next section, we define the concepts of eccentricity, radius, and diameter in stratified graphs and establish some of their basic properties. Section 3.3 focuses on the concept of centrality in stratified graphs and establishes the existence of stratified graphs with prescribed centers. In Section 3.4 we examine the concept of centrality in a properly colored tree. Section 3.5 is devoted to the concept of the periphery in stratified graphs. In the final section we introduce
some new concepts unique to stratified graphs and discuss their properties.

### 3.2 Eccentricity, Radius, and Diameter

Let $G$ be a $k$-stratified graph. The *distance* $d(u, v)$ between two vertices $u$ and $v$ of $G$ is the length of a shortest $u - v$ path. For a vertex $v$ of $G$, the *eccentricity* $e(v)$ is the distance between $v$ and a vertex furthest from $v$. For a color $X$ (one of the $k$ colors), the *$X$-eccentricity* $e_X(v)$ of $v$ is the distance between $v$ and an $X$-colored vertex furthest from $v$. So if $v$ is the unique $X$-colored vertex of $G$, then $e_X(v) = 0$. The minimum $X$-eccentricity among the vertices of $G$ is the *$X$-radius* $rad_X(G)$ of $G$, while the maximum $X$-eccentricity is the *$X$-diameter* $diam_X(G)$. Figure 20 shows a 2-stratified graph $G$ whose vertices are therefore colored red (R) and blue (B). The vertices of $G$ are labeled with their red eccentricities. Consequently, $rad_R(G) = 2$ and $diam_R(G) = 5$. Hence, note that while $rad(G) \leq diam(G) \leq 2rad(G)$ for the ordinary radius and diameter in a graph, such is not the case for $rad_R(G)$ and $diam_R(G)$ in 2-stratified graphs or, in general, for $rad_X(G)$ and $diam_X(G)$ for a color $X$ in a $k$-stratified graph $G$, as we now show.

![Figure 20](image.png)

**Theorem 12** Let $a$ and $b$ be positive integers with $a \leq b$. There exists a $k$-stratified graph $G$ and a color $X$ in $G$ such that $rad_X(G) = a$ and $diam_X(G) = b$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
**Proof:** Suppose first that $a$ and $b$ are positive integers such that $a \leq b < 2a$. If $a = b = 1$, then let $G = K_{k+1}$, where two vertices are colored $X$ and there is one vertex with each of the remaining $k-1$ colors. Then $rad_X(G) = diam_X(G) = 1$. For $a \geq 2$, we begin by identifying an end-vertex of the path $P_{b-a+1}$ with a vertex of $C_{2a}$, denoting the resulting graph by $G$. We assign each vertex of $C_{2a}$ the color $X$ as well as the end-vertex of $G$. The remaining vertices of $G$ are then colored arbitrarily with the remaining $k-1$ colors so that each color has been used. If $k$ is too large to do this, then pendant edges are added to $G$ at its vertex of degree 3 and these new vertices are colored with the remaining colors. The resulting graph $G'$ has $X$-radius $a$ and $X$-diameter $b$. This construction is illustrated in Figure 21 for $a = 4, b = 7, and k = 5$.

![Figure 21](image)

Figure 21. A 5-stratified Graph With $X$-radius 4 and $X$-diameter 7.

Next, suppose that $a$ and $b$ are positive integers with $b \geq 2a$. Define $G$ to be the path $P_{b+1}$ of order $b+1$, say $P_b : v_0, v_1, \ldots, v_b$. The vertices $v_0$ and $v_{2a}$ are assigned the color $X$, and the remaining $k-1$ colors are distributed among the remaining vertices. If $k$ is too large to do this, then pendant edges are added to $G$ at $v_{2a-1}$ and the new vertices are colored with the remaining colors. The resulting graph $G'$ has $X$-radius $a$ and $X$-diameter $b$.

This construction is illustrated in Figure 22 for $a = 3, b = 8, and k = 9$. □

For a vertex $u$ in a $k$-stratified graph $G$, we refer to the $k$-vector $\vec{e}(u) = (e_1(u), e_2(u), \ldots, e_k(u))$, where $e_i(u) = e_{X_i}(u)$, as the eccentricity vector of $u$. Note

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 22. A 9-stratified Graph With $X$-radius 3 and $X$-diameter 8.

that $e(u) = \max_i \{e_i(u) | i = 1, 2, \ldots, k\}$. The sum of $X_i$-eccentricities of $u$ is called its total eccentricity and is denoted by $\text{total}(\bar{e}(u)) = \sum_{i=1}^{k} e_i(u)$. We note that there exist $k$-stratified graphs in which the total eccentricity is the same for all vertices of $G$, for example, a 2-stratified cycle of even length in which every other vertex on the cycle belongs to the same class. Another example is a $k$-stratified cycle of length $k$. Also, in a $k$-stratified graph $G$, note that $\text{total}(\bar{e}(v)) \leq ke(v)$, for all $v \in V(G)$.

We now establish some basic results concerning $X$-eccentricities of vertices in a $k$-stratified graph.

**Theorem 13** Let $X$ be a color in a connected $k$-stratified graph $G$. If $uv \in E(G)$, then $|e_X(u) - e_X(v)| \leq 1$.

**Proof:** Suppose that $e_X(v) \leq e_X(u) = t$. We show that $e_X(v) \geq t - 1$, which will then complete the proof. Let $w$ be an $X$-colored vertex such that $e_X(u) = d(u, w)$. Thus

$$t = d(u, w) \leq d(u, v) + d(v, w) \leq 1 + e_X(v). \square$$

We now state two immediate consequences of Theorem 13.

**Corollary 2** Let $G$ be a connected $k$-stratified graph and let $t$ be an integer such that $\text{rad}_X(G) \leq t \leq \text{diam}_X(G)$. Then there exists a vertex $v$ of $G$ such that $e_X(v) = t$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Let $G$ be a connected $k$-stratified graph with color $X$. The $X$-eccentricity set is the set of $X$-eccentricities of the vertices of $G$.

**Corollary 3** Let $G$ be a connected $k$-stratified graph with color $X$ such that $\text{rad}_X(G) = r$ and $\text{diam}_X(G) = d$. Then the $X$-eccentricity set of $G$ is $\{r, r + 1, \ldots, d\}$.

**Corollary 4** Let $S = \{r, r + 1, \ldots, d\}$ be a set of positive integers with $r \leq d$. Then $S$ is the $X$-eccentricity set of some $k$-stratified graph.

For a connected graph $G$ and an integer $t$ with $\text{rad}(G) < t \leq \text{diam}(G)$, Lesniak [L.75] showed that there are, in fact, at least two vertices of $G$ having eccentricity $t$. The corresponding statement for stratified graphs is not true, however; that is, if $G$ is a $k$-stratified graph and $t$ is an integer such that $\text{rad}_X(G) < t \leq \text{diam}_X(G)$, then $G$ need not contain two vertices having $X$-eccentricity $t$ (although by Corollary 2 there must be at least one such vertex). This is illustrated in the 2-stratified graph $G$ of Figure 23 where $\text{rad}_X(G) = 3$ and $\text{diam}_X(G) = 9$ but there is only one vertex having $X$-eccentricity 7, 8, or 9.

![Figure 23. A Stratified Graph Having Only One Vertex of a Given X-eccentricity.](image)

Let $G$ be a $k$-stratified graph (with $X$ one of its colors) and define

$$s_X = \min\{e_X(v) | v \in V(G), \text{color}(v) = X\}.$$ 

So, for example, $s = 6$ in the stratified graph of Figure 23. We now show that there is an analogous result to Lesniak's theorem if $\text{diam}_X(G)$ is replaced by $s_X$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Theorem 14 If \( \text{rad}_X(G) < t \leq s_X \), then there exist at least two vertices with \( X \)-eccentricity \( t \).

**Proof:** Let \( u \) be a vertex with \( e_X(u) = t \). There is an \( X \)-colored vertex \( v \) with \( d(u, v) = t \). Let \( w \) be a vertex such that \( e_X(w) = \text{rad}_X(G) \). Let \( P \) be a \( w - v \) path in \( G \) of length \( d(w, v) \). Then \( e_X(w) = \text{rad}_X(G) \) and \( e_X(v) \geq s_X \). But \( \text{rad}_X(G) < t \leq s_X \leq e_X(v) \). So on \( P \) there is a vertex \( x \) such that \( e_X(x) = t \). We show that \( u \neq x \).

\[
d(u, v) = t > \text{rad}_X(G) \geq d(w, v) > d(x, v)
\]

this implies that, \( d(u, v) > d(x, v) \). Therefore, \( u \neq x \). \( \square \)

3.3 Centers in Stratified Graphs

The center \( C(G) \) of a connected graph \( G \) is the subgraph of \( G \) induced by those vertices with eccentricity \( \text{rad}(G) \). In this section we introduce and investigate the corresponding concept for stratified graphs. Let \( X \) be a color in a \( k \)-stratified graph. The \( X \)-center \( C_X(G) \) of \( G \) is the subgraph induced by those vertices \( v \) with \( e_X(v) = \text{rad}_X(G) \). This is illustrated for the three 2-stratified graphs \( G_1, G_2, \) and \( G_3 \) of Figure 24, where in each case, the vertices of the 2-stratified graphs are labeled with their red eccentricities.

Note that in the 2-stratified graph \( G_2 \) of Figure 24, the red center is also the blue center. However, in the 2-stratified graph \( G_3 \) of the same figure, the blue center is a proper subgraph of the red center. Figure 25 provides an example in which both the red center \( C_R(G) \) and the blue center \( C_B(G) \) are disconnected. Each vertex \( v \) of \( G \) is also labeled with the ordered pair \( (e_R(v), e_B(v)) \).

In this example as well as in \( G_2 \) and \( G_3 \) of Figure 24 the union of the red
Figure 24. Three 2-stratified Graphs and Their Red Centers.

center and the blue center is the ordinary center of $G$. This is not true in general as Figure 26 shows. However, in Figure 26, $\langle C_R(G) \cap C_B(G) \rangle = C(G)$.

A well-known property of the center of a connected graph $G$ is that it always lies in a single block of $G$. The same is true for $k$-stratified graphs.

**Theorem 15** Let $G$ be a connected $k$-stratified graph, $k \geq 1$. For each color $X$ of $G$, the $X$-center $C_X(G)$ lies in a single block of $G$.

**Proof:** Suppose that this is not the case. Then $G$ contains a cut-vertex $v$ of $G$ such that distinct components of $G - v$ contain vertices of $C_X(G)$. Let $u$ be an $X$-colored vertex of $G$ such that $e_X(v) = d(u, v)$. Let $w$ be a vertex of $C_X(G)$ belonging to a component of $G - v$ distinct from that containing $u$. Then $rad_X(G) = e_X(w) \geq d(w, u) = d(w, v) + d(v, u) > e_X(v)$. Thus $e_X(v) < rad_X(G)$, which is a contradiction. □

We now have an analogue of a well-known result on trees.

**Corollary 5** For each $k$-stratified tree $T$ and each color $X$, the $X$-center $C_X(T) \cong K_1$ or $C_X(T) \cong K_2$.
Figure 25. $C_R(G)$ and $C_B(G)$ Are Disconnected. $\langle C_R(G) \cup C_B(G) \rangle = C(G)$.

Hedetniemi (see [FMP81]) showed that every graph is the center of some connected graph. Using his proof technique, we establish the corresponding result for stratified graphs.

Theorem 16 For every $k$-stratified graph $H$, there exists a $k$-stratified graph $G$ such that $C_X(G) = H$.

Proof: Since the result is true for $k = 1$, we assume that $k \geq 2$. To construct $G$, we add four vertices $u_1, v_1, u_2, v_2$ to $H$ and join $v_i$ ($i = 1, 2$) to all vertices of $H$ as well as add the edges $u_i v_i$ ($i = 1, 2$). The vertices $u_1, v_1, u_2, v_2$ are all colored $X$. Then $e_X(u_i) = 4$ and $e_X(v_i) = 3$ for $i = 1, 2$; while $e_X(w) = 2$ for all vertices $w$ of $H$. Thus $C_X(G) = \langle V(H) \rangle = H$. See Figure 27 for an illustration of this construction. □

We now show that it is possible to prescribe all $k$ centers of a $k$-stratified graph simultaneously.
Figure 26. \((C_R(G) \cap C_B(G)) = C(G)\).

Figure 27. \(k\)-stratified Graph \(G\) With \(X\)-center \(C_X(G) = H\).

**Theorem 17** For \(k\)-stratified graphs \(H_1, H_2, \ldots, H_k\) with colors \(X_1, X_2, \ldots, X_k\), there exists a \(k\)-stratified graph \(G\) such that \(C_{X_i}(G) \cong H_i\) for \(i = 1, 2, \ldots, k\).

**Proof:** We begin with a copy of each of the \(k\)-stratified graphs \(H_1, H_2, \ldots, H_k\). Then add all edges of the form \(v_i v_j\), where \(v_i \in V(H_i), v_j \in V(H_j)\) and \(i \neq j\). Next for each \(k\)-stratified graph \(H_i\), \(i = 1, 2, \ldots, k\), add one vertex \(u_i\) and join \(u_i\) to every vertex of \(H_i\). Color \(u_i\) with color \(X_{i+1}\), where \(X_{k+1} = X_1\). Finally, for each \(i = 1, 2, \ldots, k\), add a vertex \(w_i\) colored \(X_i\), and the edge \(w_i u_i\). Denote the resulting \(k\)-stratified graph by \(G\). (See Figure 28.)

Observe that \(e_{X_i}(w_1) = 5\) and \(e_{X_i}(w_i) = 4\) for \(i \neq 1\). Also, \(e_{X_i}(u_1) = 3\) and \(e_{X_i}(u_i) = 4\) for \(i \neq 1\). Moreover, if \(v \in V(H_1)\), then \(e_{X_i}(v) = 2\) while, for \(v \in V(H_i), i \neq 1\), we have \(e_{X_i}(v) = 3\). Therefore, \(C_{X_i}(G) \cong H_1\). Similarly,
Figure 28. Construction of a \( k \)-stratified Graph \( G \) With \( k \) Prescribed Centers.

\[ C_{X_i}(G) \cong H_i \text{ for } 1 \leq i \leq k. \]

The distance of two subgraphs \( G_1 \) and \( G_2 \) of a graph \( G \) is defined by

\[ d(G_1, G_2) = \min \{ d(v_1, v_2) \mid v_1 \in V(G_1), v_2 \in V(G_2) \}. \]

In the proof of Theorem 17, the distance between \( C_{X_i}(G) \) and \( C_{X_j}(G), i \neq j \), is 1. We next show that this condition is not needed. We can extend this result further by requiring every pair of centers to be arbitrarily far apart.

**Theorem 18** For \( k \)-stratified graphs \( H_1, H_2, \ldots, H_k \) and every integer \( n \geq 2 \), there exists a \( k \)-stratified graph \( G \) such that \( C_{X_i}(G) \cong H_i \) for \( i = 1, 2, \ldots, k \) and
Proof: We generalize the construction of the graph in the proof of Theorem 17. For each integer \( i \) (1 \( \leq i \leq k \)), add a vertex \( z_i \) and all edges of the form \( z_i v \) for all \( v \in V(H) \). Further, for distinct integers \( i, j \) with \( 1 \leq i, j \leq k \), the vertices \( z_i \) and \( z_j \) are connected by a path of length \( n - 2 \). The colors of these added vertices is chosen arbitrarily. Next for each \( i \) (1 \( \leq i \leq k \)) we add a vertex \( u_i \) colored \( X_{i+1} \) (where \( X_{k+1} = X \)), and all edges \( u_i v \) for all \( v \in V(H) \). Finally adjoin a path of length \( n \) to \( u_i \), where each vertex on the path is colored \( X_i \). (See Figure 29 for an example with \( k = 3 \) and \( n = 4 \).) This completes the construction of the \( k \)-stratified graph \( G \), which then has the desired properties. \( \square \)

3.4 Centers in Properly 2-Colored 2-Stratified Trees

In this section we investigate the impact of coloring patterns, specifically that of a proper 2-coloring on the structure of the center of a tree. For a graph \( G \), any path in \( G \) of length \( \text{diam}(G) \) is called a diametrical path in \( G \). It is useful to recall that for every tree \( T \), either \( \text{diam}(T) = 2 \text{rad}(T) \) or \( \text{diam}(T) = 2 \text{rad}(T) - 1 \) according to whether \( \text{diam}(T) \) is even or odd, respectively. Also, observe that \( C(T) \cong K_1 \) if and only if \( \text{diam}(T) = 2 \text{rad}(T) \), and \( C(T) \cong K_2 \) if and only if \( \text{diam}(T) = 2 \text{rad}(T) - 1 \). The following basic properties given in the next two lemmas are of general knowledge, but we prove them for completeness.

Lemma 5 The center of a tree \( T \) is a subgraph of every diametrical path of \( T \).

Proof: Let \( T \) be a tree and let \( P \) be a diametrical path in \( T \). Suppose that \( P \) is a \( u - v \) path, where, necessarily, \( u \) and \( v \) are end-vertices of \( T \). Now, suppose, to the contrary, that there is a central vertex \( w \) of \( T \) not on \( P \). Let \( x \) denote the
vertex on $P$ closest to $w$. Therefore, $d(w, x) \leq d(w, y)$ for all $y \in V(P)$. Since $w$ is not on $P$ but $x$ is, $d(w, x) > 0$.

Observe that the $w - v$ path and $w - u$ path in $T$ must pass through $x$. Thus for a central vertex $w$ of $T$, it follows that

$$rad(T) = e(w) \geq d(w, u) = d(w, x) + d(x, u)$$

(6)

$$rad(T) = e(w) \geq d(w, v) = d(w, x) + d(x, v)$$

(7)

Now (6) and (7) imply that $2 \cdot rad(T) \geq 2 \cdot d(w, x) + d(x, u) + d(x, v)$. But, $d(x, u) + d(x, v) = diam(T)$. Hence, $2 \cdot rad(T) \geq 2 \cdot d(w, x) + diam(T)$. Whether
diam(T) = 2 rad(T) or diam(T) = 2 rad(T) − 1, a contradiction is produced. □

Lemma 6 Let T be a tree and let P denote a u − v diametrical path of T. If w is a central vertex of T, then rad(T) = e(w) = max{d(u, w), d(w, v)}

Proof: Let P denote a diametrical u − v path of T. By Lemma 5, w is on P. Clearly, e(w) ≥ max{d(u, w), d(w, v)}. Now, we want to show that e(w) = max{d(u, w), d(w, v)}. So, assume to the contrary that e(w) > max{d(u, w), d(w, v)}. Then there is a vertex x in T such that x is not on P and e(w) = d(w, x). Let w = w₀, w₁, .. , wₜ = x be the shortest w − x path in T, and let j denote the largest positive integer for which wⱼ is on P. Assume, without loss of generality, that d(u, wⱼ) ≥ d(wⱼ, v). Observe that d(wⱼ, x) > d(wⱼ, v) because, otherwise, d(wⱼ, x) + d(wⱼ, v) ≤ d(w, wⱼ) + d(wⱼ, v). But this implies that d(w, x) ≤ d(w, v), which contradicts the assumption that e(w) = d(w, x) > max{d(u, w), d(w, v)}. Therefore, d(u, x) = d(u, wⱼ) + d(wⱼ, x) > d(u, wⱼ) + d(wⱼ, v) = d(u, v), which contradicts the fact that d(u, v) is the length of a diametrical path of T. □

Corollary 6 Let T be a tree with unique central vertex w. If P is a u − v diametrical path of T, then d(u, w) = d(w, v).

Proof: By the previous lemma, e(w) = max{d(u, w), d(w, v)}. Since C(T) ≅ K₁, diam(T) = 2 rad(T). But rad(T) = max{d(u, w), d(w, v)} and diam(T) = d(u, w) + d(w, v). Without loss of generality, assume that d(u, w) = max{d(u, w), d(w, v)}. Then 2d(u, w) = d(u, w) + d(w, v). It follows that d(u, w) = d(w, v). □

Corollary 7 Let T be a tree. If C(T) ≅ K₂ and P is a diametrical u − v path of T, then for a central vertex w of T,

max{d(u, w), d(w, v)} − min{d(u, w), d(w, v)} = 1.
**Proof:** Without loss of generality, assume that $d(u, w) = \max\{d(u, w), d(w, v)\}$. Hence, $\text{rad}(T) = e(w) = d(u, w)$. But,

$$
diam(T) = 2 \text{rad}(T) - 1
= 2 e(w) - 1
= 2 d(u, w) - 1
$$

Hence, $diam(T) = d(u, w) + d(w, v) = 2 d((u, w) - 1$. Thus, $d(w, v) = d(u, w) - 1$. Therefore, $d(u, w) - d(w, v) = 1$. □

The following lemma is a consequence of the fact that 2-colored graphs are bipartite.

**Lemma 7** Let $G$ be a properly 2-colored 2-stratified graph. For $u, v \in V(G)$, $\text{color}(u) = \text{color}(v)$ if and only if $d(u, v)$ is even.

**Corollary 8** Let $G$ be a properly 2-colored 2-stratified graph with vertex $u$ and color $X$. Then $e_X(u)$ is even if and only if $u$ is colored $X$.

**Proof:** Let $v$ be an $X$-colored vertex of $G$ such that $d(u, v) = e_X(u)$. Then $e_X(u)$ is even if and only if $\text{color}(u) = \text{color}(v) = X$. □

Recall, for each $u \in V(T)$, that $\text{total}(\tilde{e}(u))$ is the sum of $X_i$-eccentricities $e_{X_i}(u)$ of $u$ for $i = 1, 2, \ldots, k$.

**Theorem 19** Let $T$ be a properly 2-colored 2-stratified tree. Then for each $u \in V(T)$, $\text{total}(\tilde{e}(u))$ is odd.

**Proof:** By Corollary 7, if $\text{color}(u) = R$, then $e_R(u)$ is even and $e_B(u)$ is odd. □

**Lemma 8** Let $T$ be a 2-stratified tree. For each $v \in V(T)$, the eccentricity $e(v) = \max\{e_R(v), e_B(v)\}$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Theorem 20 Let $T$ be a properly 2-colored 2-stratified tree. Then for each vertex $u$ in $T$, $|e_R(u) - e_B(u)| = 1$.

Proof: For each vertex $u$ in $T$, let $A_i(u) = \{w \in V(T)|d(u, w) = i\}$. Let $e_B(u) = s$. By Theorem 19, $\text{total}(\varepsilon(u))$ is odd, so, without loss of generality, assume that $e_R(u) < e_B(u) = s$. Now, we show that $e_R(u) \geq s - 1$ which completes the proof. According to Lemma 8, $e(u) = e_B(u) = s$. Also, observe that vertices in $A_s(u)$ are colored blue and those in $A_{s-1}(u)$ are colored red. Let $v \in V(T)$ such that $d(u, v) = s$. Then $v \in A_s(u)$. Consider the unique $u - v$ path $u = u_0, u_1, \ldots, u_s = v$. It follows that $u_{s-1} \in A_{s-1}(u)$. Hence, $u_{s-1}$ is colored red. Therefore, $e_R(u) \geq s - 1$. □

Theorem 21 Let $T$ be a properly 2-colored 2-stratified tree. Then for each color $X$, $C_X(T) \cong K_1$.

Proof: Without loss of generality, assume that $X = R$, and suppose, to the contrary, that $C_R(T) \cong K_2$. Let $u$ and $v$ be the vertices in $C_R(T)$. Since $T$ is properly 2-colored and $uv \in E(T)$, $\text{color}(u) \neq \text{color}(v)$. Since $u$ and $v$ are in $C_R(T)$, it follows that $e_R(u) = e_R(v)$ and they have the same parity. Thus, by Corollary 8, $\text{color}(u) = \text{color}(v) = R$ or $\text{color}(u) = \text{color}(v) = B$ according to whether $e_R(u)$ and $e_B(v)$ are both even or are both odd, respectively. This contradicts the fact that $T$ is properly 2-colored. □

Theorem 22 Let $w$ be a vertex in a properly 2-colored 2-stratified tree $T$. Then $\text{total}(\varepsilon(w)) = \text{diam}(T) - 1$ if and only if $\text{diam}(T)$ is even and $w$ is a central vertex. Also, $\text{total}(\varepsilon(w)) = \text{diam}(T)$ if and only if $\text{diam}(T)$ is odd and $w$ is a central vertex.
Proof: Assume first that $w$ is a central vertex of $T$ and that $diam(T)$ is even. Let $u$ and $v$ denote the end-vertices of a diametrical path $P$. It follows by Lemma 5 that $w$ is on $P$. Also, note that $e(w) = max\{e_R(w), e_B(w)\}$. Without loss of generality, assume that $color(u) = R$. Then $e_R(w) = e(w)$. Thus, $d(u, w) = d(w, v) = e(w) = e_R(w) = rad(T)$. However, $|e_R(w) - e_B(w)| = 1$. Therefore,

$$total(e(w)) = e_R(w) + e_B(w) = e_R(w) + (e_R(w) - 1) = 2e_R(w) - 1 = 2rad(T) - 1 = diam(T) - 1.$$ 

For the converse, assume that $total(e(w)) = diam(T) - 1$. Then, without loss of generality, assume that $e_R(w) > e_B(w)$. Therefore,

$$total(e(w)) = e_R(w) + e_B(w) = e_R(w) + (e_R(w) - 1) = 2e_R(w) - 1 = diam(T) - 1.$$ 

Therefore, $2e_R(w) = diam(T)$ and $diam(T)$ is even. So, $2e_R(w) = diam(T) = 2rad(T)$, implying that $e_R(w) = rad(T)$ and that $w$ is a central vertex of $T$.

Next, assume that $diam(T)$ is odd and $w$ is a central vertex of $T$. For any vertex $v$ of $T$ we know that

$$total(e(v)) = e_R(v) + e_B(v),$$
$$|e_R(v) - e_B(v)| = 1,$$
$$e(v) = max\{e_R(v), e_B(v)\}.$$
We show that \( \text{total}(e(w)) = \text{diam}(T) \). Again, consider a diametrical \( u - v \) path \( P \). As we have seen, \( w \) is on \( P \) and \( e(w) = \max\{e_R(w), e_B(w)\} \). Without loss of generality, assume that \( \text{color}(u) = R \) and \( e_R(w) = e(w) \), so \( e(w) = d(u, w) \). Then, \( d(u, w) - d(w, v) = 1 \), Thus, \( d(w, v) = e_R(w) - 1 = e_B(w) \). Hence, \( \text{diam}(T) = d(u, w) + d(w, v) = e_R(w) + e_B(w) = \text{total}(e(w)) \).

Now, for the converse, suppose that \( \text{total}(e(w)) = \text{diam}(T) \). Then, \( e_R(w) + e_B(w) = \text{diam}(T) \). Hence, \( \text{diam}(T) \) is odd. Thus, \( \text{diam}(T) = 2 \text{rad}(T) - 1 \).

Without loss of generality, assume that \( e_R(w) \geq e_B(w) \). So, \( \text{rad}(T) = e(w) = e_R(w) \). And, since \( T \) is properly 2-colored it follows from Theorem 20 that \( e_R(w) - e_b(w) = 1 \). Hence, \( e_B(w) = e_R(w) - 1 \). Therefore

\[
\text{total}(e(w)) = e_R(w) + e_B(w) = \text{diam}(T) = 2 \text{rad}(T) - 1 = 2 e_R(w) - 1
\]

Hence, \( e_R(w) = \text{rad}(T) \), which implies that \( w \) is a central vertex of \( T \). \( \square \)

Figure 30. The Set of Central Vertices of \( G \) Is Not a Subset of the Union of the Red and Blue Centers.

Figure 30 gives an example of a 2-stratified graph \( G \) for which \( C(G) \nsubseteq (C_R(G) \cup C_B(G)) \). This raises the question of when \( C(G) \subseteq (C_R(G) \cup C_B(G)) \).
Next we show this to be true for properly 2-colored 2-stratified trees.

**Theorem 23** Let $T$ be a properly 2-colored 2-stratified tree. Then for a color $X$, $C_X(T)$ is a subgraph of $C(T)$. More precisely if $C(T) \cong K_1$, then $C_X(T) = C(T)$ and if $C(T) \cong K_2$, then $(C_R(T) \cup C_B(T)) = C(T)$.

**Proof:** First, assume that $T$ has one central vertex, say $w$. There exists $u \in V(T)$ such that $d(u, w) = e(w)$. Assume, without loss of generality, that $color(u) = red$. Then $e_R(w) = e(w)$. Since by Theorem 21 $C_R(T) \cong K_1$, it follows that $w$ is the only vertex of $C_R(T)$. Now, we show that $w$ is also the only vertex of $C_B(T)$. By Theorem 20, $e_B(w) = e(w) - 1$. By Lemma 5, the central vertex $w$ is on every diametrical path. And, by Lemma 6, the end-vertex $u$ is on some diametrical path. Let $P : u = u_0, u_1, \ldots, u_t = v$ denote such a path. Observe that $color(u_{t-1})$ is blue and that $d(w, u_{t-1}) = e(w) - 1 = e_B(w)$. Therefore, $w$ is also in $C_B(T)$.

Now, let $C(T) \cong K_2$. By Theorem 21 $C_R(T) \cong K_1$ and also $C_B(T) \cong K_1$. Let $w_1$ denote a central vertex of $T$. By Lemma 5, $w_1$ is on every diametrical path of $T$. If $v$ is a vertex of $T$ such that $e(w_1) = d(w_1, v)$, then by Lemma 6, $v$ is on some diametrical $u-v$ path $P$ of $T$. So $d(w, v) = \operatorname{rad}(T)$ and $d(w, u) = \operatorname{rad}(T) - 1$. Without loss of generality, assume that $color(v) = red$. Hence, $w_1$ is the red center of $T$. Also, note that $d(u, v)$ is odd. Hence, $color(v) \neq color(u)$. Thus, $color(u) = blue$. Let $w_2$ be the vertex that follows $w$ on $P$ as we proceed from $u$ to $v$. Then $w_2$ is the other central vertex of $T$, and $\operatorname{rad}(T) = e(w_2) = d(w_2, u)$. Since $u$ is blue, $e_B(w_2) = e(w_2) = \operatorname{rad}(T)$, which says that $w_2$ is the blue center of $T$. □

We have an immediate corollary.

**Corollary 9** Let $T$ be a properly 2-colored 2-stratified tree. For each color $X$, $C_X(T)$ is a subgraph of every diametrical path of $T$. 

Republished with permission of the copyright owner. Further reproduction prohibited without permission.
3.5 Periphery

The concept that is opposite to the center of a graph is its periphery. The periphery $P(G)$ of a connected graph is the subgraph of $G$ induced by those vertices $v$ with $e(v) = diam(G)$. Bielak and Syslo [HM83] showed that a graph $G$ is the periphery of a connected graph if and only if every vertex of $G$ has eccentricity 1 or no vertex of $G$ has eccentricity 1.

For a color $X$ of a $k$-stratified graph $G$, the $X$-periphery $P_X(G)$ is the subgraph of $G$ induced by those vertices $v$ with $e_X(v) = diam_X(G)$. For the 2-stratified graph $G$ of Figure 31 the red eccentricity of each vertex is shown along with the red periphery.

We now present a characterization of those $k$-stratified graphs that are the $X$-periphery of some $k$-stratified graph.

**Theorem 24** Let $G$ be a $k$-stratified graph with color $X$. Then $G$ is the $X$-
periphery of a \( k \)-stratified graph if and only if every vertex of \( G \) has \( X \)-eccentricity 1 or no \( X \)-colored vertex of \( G \) has \( X \)-eccentricity 1.

**Proof:** If every vertex of \( G \) has \( X \)-eccentricity 1, then \( G \) is the \( X \)-periphery of \( G \) itself. Suppose, next, that \( G \) is a \( k \)-stratified graph in which no \( X \)-colored vertex of \( G \) has \( X \)-eccentricity 1. In this case, define \( H \) to be that \( k \)-stratified graph obtained by adding an \( X \)-colored vertex \( v \) to \( G \) and joining \( v \) to all vertices of \( G \). Then \( v \) has \( X \)-eccentricity 1 and every vertex of \( G \) has \( X \)-eccentricity 2, so \( G \) is the \( X \)-periphery of \( H \).

For the converse, suppose, to the contrary, that there is a \( k \)-stratified graph \( G \) for which some \( X \)-colored vertex \( u \) has \( X \)-eccentricity 1 but not all vertices of \( G \) have \( X \)-eccentricity 1 such that \( G \) is the \( X \)-periphery of a \( k \)-stratified graph \( H \). Since not all vertices of \( G \) have the same \( X \)-eccentricity, \( G \) is a proper induced subgraph of \( H \). Suppose that \( \text{diam}_X(H) = k \), where, then, \( k \geq 2 \). In \( H \), \( e_X(u) = k \).

Hence there exists an \( X \)-colored vertex \( w \) such that \( d(u, w) = k \). Since in \( G \), \( e_X(u) = 1 \), it follows that \( u \) is adjacent to all other \( X \)-colored vertices of \( G \). This implies that \( w \) does not belong to \( G \). However, since \( d(w, u) = k \) and \( u \) is \( X \)-colored, it follows that \( e_X(w) = k \) and so \( w \) belongs to the \( X \)-periphery of \( H \), but this is a contradiction. \( \Box \)

### 3.6 Proximity, Maximum Proximity, Seclusion

For a vertex \( v \) in a \( k \)-stratified graph \( G \), there are situations where it is of interest to know the minimum distance between \( v \) and a vertex in some prescribed stratum. It is this fact that leads us to the concepts in this section.

For a vertex \( v \) of a \( k \)-stratified graph \( G \) and a color \( X \) of \( G \), the **\( X \)-proximity** \( \delta_X(v) \) of \( v \) is the distance between \( v \) and an \( X \)-colored vertex closest to \( v \). If \( v \)
itself is an $X$-colored vertex, then $\delta_X(v) = 0$. This concept is, in a sense, opposite to that of the $X$-eccentricity.

If the vertices of a graph are not partitioned, such as in a 1-stratified (or ordinary) graph, then the proximity of every vertex is 0. Hence, in the domain of ordinary graphs, proximity is not an interesting concept for study. A basic result regarding the $X$-proximity of adjacent vertices in a $k$-stratified graph $G$ is given below. It parallels Theorem 13.

**Theorem 25** Let $X$ be a color in a connected $k$-stratified graph. If $uv \in E(G)$, then $|\delta_X(u) - \delta_X(v)| \leq 1$.

**Proof:** Suppose that $\delta_X(u) \geq \delta_X(v) = t$. We show that $\delta_X(u) \leq t + 1$, which will then complete the proof. Let $w$ be the closest $X$-colored vertex to $v$. So $d(v, w) = t$. Thus

$$\delta_X(u) \leq d(u, w) \leq d(u, v) + d(v, w) \leq 1 + t. \quad \square$$

For a color $X$, the maximum $X$-proximity $\Delta_X(G)$ is the greatest $X$-proximity among all vertices of $G$. This concept is analogous to that of diameter in graphs. The minimum $X$-proximity among all vertices of $G$ is always 0, and this value is attained by all $X$-colored vertices in $G$. We define the $X$-seclusion $S_X(G)$ of $G$ to be the subgraph induced by those vertices $v$ of $G$ with $\delta_X(v) = \Delta_X(G)$. For the 2-stratified graph of Figure 32 the red proximity of each vertex is shown along with the red seclusion $S_R(G)$.

**Theorem 26** For positive integers $a$ and $b$, there exists a 2-stratified graph with $\Delta_R(G) = a$ and $\Delta_B(G) = b$.  

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 32. A 2-stratified Graph \( G \) and Its Red Seclusion.

**Proof:** Consider a path of order \( a + b \) such that the first \( a \) vertices are colored red and the remaining \( b \) vertices are colored blue. In this path one end vertex is colored red and the other is colored blue. The distance from the end vertex to the closest vertex of the other color is \( a \) and \( b \), respectively. Figure 33 illustrates a 2-stratified path with \( \Delta_B(G) = 5 \) and \( \Delta_R(G) = 8 \).

This gives rise to the problem of determining those integers \( t_i \) (\( 1 \leq i \leq k \)) for which there exist a \( k \)-stratified graph with \( \Delta_{X_i}(G) = t_i \).

We now turn to the problem of determining which graphs are \( X \)-seclusions.

**Theorem 27** For every \( l \)-stratified graph \( H \) and every integer \( k \) where \( k > l \), there exists a \( k \)-stratified graph \( G \) and a color \( X \) of \( G \) that is not a color of \( H \) such that \( S_X(G) = H \).

**Proof:** If \( X \) is a color of a \( k \)-stratified graph \( G \) (\( k \geq 2 \)), then, necessarily, no vertex of \( S_X(G) \) is \( X \)-colored, that is, \( S_X(G) \) is \( l \)-stratified for \( l < k \). Let \( H \) be an \( l \)-stratified graph, let \( k \) be an integer with \( k > l \), and let \( X_1, X_2, \ldots, X_{k-l} = X \)
be $k - l$ colors not used in $H$. Further, let $P : v_1, v_2, \ldots, v_{k-l}$ be a path of order $k - l$ where $v_i$ is colored $X_i$. We construct $G$ from $H$ and $P$ by joining $v_1$ to every vertex of $H$. Then in $G$, $\delta_X(v) = k - l$ for every vertex $v$ in $H$; while $\delta_X(v_i) = k - l - i$ for $1 \leq i \leq k - l$. Thus $\Delta_X(G) = k - l$ and $S_X(G) = H$. See Figure 34 for an illustration of this construction.

![Diagram](image)

Figure 34. The $l$-stratified Subgraph $H$ is the $X$-seclusion of the $k$-stratified Graph $G$.

**Corollary 10** Every $k$-stratified graph $G$ is the $X$-seclusion of some $(k + 1)$-stratified graph $H$.

**Proof:** Add a vertex $u$ to $G$. Assign color $X$ to $u$ and join $u$ to every vertex of $G$. The $X$-seclusion of $H$ is $G$. □
CHAPTER IV

ALGORITHMS IN STRATIFIED GRAPHS

4.1 Introduction

In this chapter we introduce results concerning stratified graphs that are algorithmic in nature. In particular, we describe algorithms solving problems dealing with stratified graphs and present an analysis of their complexity. If an efficient algorithm is not found, we investigate whether the problem is NP-complete.

Many known algorithms about graphs can be used to solve similar problems involving stratified graphs. We shall mention briefly a number of these. Other algorithms will be discussed in more detail.

In the next section we concentrate on problems concerned with finding subgraphs that completely avoid, minimize, or maximize the number of vertices of a certain color. In Section 4.3 we turn our attention to the question of how to find most stratified (colorful) subgraphs.

4.2 Finding Optimized Subgraphs

First we consider the case in which a certain color in a stratified graph is to be avoided. This condition is then relaxed so that we find a subgraph that minimizes the number of vertices of a certain color. A shortest path problem and a variant of the spanning tree problem are considered next. In certain situations we may be interested in an induced subgraph of a $k$-stratified graph such that the vertices of the subgraph belong to the same stratum. As an example, consider the
problem of finding the longest unicolored path in $G$. We formally state a decision version of this problem.

**LONGEST UNICOLORED PATH**

**INSTANCE:** Given a $k$-stratified graph $G$ and a positive integer $M$.

**QUESTION:** Does there exist a unicolored path in $G$ of order at least $M$?

A polynomial algorithm for this problem yields a polynomial solution to the well-known longest path problem [MD79] in graphs, which is known to be NP-complete. Hence, we have the following result.

**Theorem 28** The longest unicolored path problem in a $k$-stratified graph $G$ is NP-hard.

However, observe that it is straightforward to find a unicolored acyclic subgraph of maximum order. This can be done by first finding a unicolored connected subgraph of maximum order and then finding a minimum spanning tree of this subgraph.

Next we present an algorithm for finding an induced unicolored connected subgraph of maximum order in a $k$-stratified graph $G$. In this algorithm, $Adj[G]$ refers to the adjacency matrix of $G$. We write $H_i$ and $Adj[H_i]$ for the $X_i$-colored subgraph of $G$ and its adjacency matrix, respectively. We denote a connected component of maximum order in $H_i$ by $H_i'$.

It is straightforward to verify that this algorithm correctly finds the largest induced unicolored connected subgraph of $G$ in a time bounded by a quadratic polynomial of the order of $G$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Algorithm Find_Unicolored

Input: Graph $G = (V, E), V = S_1 \cup S_2 \cup \ldots \cup S_k$.
Output: A maximum order unicolored subgraph $H$ of $G$.

Find_Unicolored($G = (V, E), V = S_1 \cup S_2 \cup \ldots \cup S_k$)
1. $H \leftarrow \emptyset$.
2. For $i$ from 1 to $k$ do;
3. For each color $X \neq X_i$ of $G$ do;
5. For each $v \in V(G)$ with $\text{color}(v) = X$ do;
6. $Adj[H_i] \leftarrow Adj[H_i] - row(v) - col(v).$
   \{Erase the row and column corresponding to $v$.\}
7. $H'_i \leftarrow$ largest component of $Adj[H_i]$.
8. If $\text{order}(H) < \text{order}(H'_i)$ then $H \leftarrow H'_i$.

Figure 35. Algorithm Find_Unicolored.

Theorem 29 Given a $k$-stratified graph $G$, the largest induced unicolored connected subgraph of $G$ can be found in $O(n^2)$ time.

Now we turn our attention to finding subgraphs that are not unicolored but are, in a sense, close to this. We begin by looking for paths with this property. For a given color $X$ in a $k$-stratified graph $G$, we are interested in finding a $u - v$ path with as many $X$-colored vertices and as few other colors as possible. This problem is primarily of interest only in 2-stratified graphs because any similar problem for a $k$-stratified graph $G$ with $k > 2$ can be mapped into an equivalent question for a 2-stratified graph $H$, where $H \cong G$ and $V(H)$ has the stratification $S_1 = \{v \in V(G)|v \text{ is colored } X \text{ in } G\}$ and $S_2 = V(G) - S_1$. Clearly, a shortest $u - v$ path in $H$ with a maximum number of vertices from $S_1$ corresponds to a
$u - v$ path in $G$ that maximizes the number of $X$-colored vertices.

Hence, we are concerned here with the problem of given a 2-stratified graph $G$ with a pair of vertices $u$ and $v$, find a shortest $u - v$ path in $G$ containing a minimum number of vertices of a certain color. This is a variant of the shortest path problem in the presence of obstacles considered in the literature. We describe one application for this problem. In a war-torn region, a shipment of medicine and supplies is to be sent from a city $A$ to a destination under siege by enemy forces. Given a map of the region, where enemy concentration points and the destinations between every pair of adjacent intersections are clearly marked, how can we find a shortest route that passes through the minimum number of enemy concentration points?

A well-known greedy algorithm due to Dijkstra [E.W59], provides an efficient solution to this problem for a weighted directed graph, where all edge weights are nonnegative. We provide a transformation that converts a given 2-stratified graph $G = (V,E)$ into a corresponding weighted directed graph $D = (V,A)$. We show that under this transformation, for vertices $u$ and $v$ of $D$, a shortest $u - v$ path found by Dijkstra's algorithm in $D$ corresponds to a shortest $u - v$ path in $G$ with a minimum number of vertices all having the same prescribed color, say blue.

For a given $k$-stratified graph $G = (V,E)$ of order $n$ with stratification $V = S_1 \cup S_2 \cup \ldots \cup S_k$, define a weighted directed graph $D = (V,A)$ such that $V(D) = V(G)$ and for each $uv \in E(G)$ there is a pair $(u,v)$, and $(v,u)$ of arcs in $E(A)$. Further, assign weights $w$ to arcs of $A$ so that if $\text{color}(v) = \text{red}$, then $w(u,v) = 1$. Also, if $\text{color}(v) = \text{blue}$, then $w(u,v) = n$. See Figure 36 for an illustration of this process. According to our construction, the weights of the arcs of $D$ are either 1 or 15, depending on whether $\text{color}(v)$ is red or blue,
Figure 36. A 2-stratified Graph $G$ With Corresponding Weighted Digraph $D$.

respectively. For clarity, in Figure 36, only those arcs with weight 15 are labeled. Observe that there are two $u-v$ paths in $G$ with only one blue vertex, namely, $P_1 : u, a, b, c, d, e, v$ and $P_2 : u, j, k, l, m, v$. But, since the length of $P_1$ in $D$ is 20, while the length of $P_2$ is only 19, Dijkstra’s algorithm selects $P_2$, which corresponds to the correct solution in $G$. In general, since the cost of passing through a blue vertex $n$ is equal to the number of vertices in $D$ and the cost of passing through a red vertex is only 1, it is cheaper to pass through all red vertices than passing through a single blue vertex. Table 3 illustrates the application of Dijkstra’s algorithm to the weighted directed graph of Figure 36.
<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>m</th>
<th>l</th>
<th>y</th>
<th>f</th>
<th>i</th>
<th>y</th>
<th>b</th>
<th>f</th>
<th>e</th>
<th>p</th>
<th>e</th>
<th>q</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>t</td>
<td>(w'61)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>(w'61)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m</td>
<td>(w'61)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>(−'oo)</td>
<td>(t'81)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>(−'oo)</td>
<td>(t'81)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(y'71)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(y'71)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(y'71)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(y'71)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>q</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(f'2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(f'2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(−'oo)</td>
<td>(n'1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Extended to

Application of Dijkstra's Algorithm to Digraph in Figure 36

Table 3

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Next, we consider a variant of the minimum spanning tree problem in graphs. Given a \( k \)-stratified graph \( G \) and a color \( X \), we are interested in finding a minimum order tree that spans all the \( X \)-colored vertices in \( G \). First, we describe the Steiner Tree Problem in graphs.

**STEINER TREE IN GRAPHS (ST)**

**INSTANCE:** Given a graph \( G = (V, E) \), a subset \( R \subseteq V \) and a positive integer \( M \leq |V| - 1 \).

**QUESTION:** Is there a subtree of \( G \) that includes all the vertices of \( R \) and that contains no more than \( M \) edges?

The STEINER TREE IN GRAPHS is known to be NP-complete [R.M72]. Observe that there is a trivial transformation from ST to the decision version of the problem that we are interested in. Namely, let \( color(v) = X \) if \( v \in R \) and arbitrarily choose a color for each \( u \in V - R \). hence, we have the following.

**Theorem 30** Given a \( k \)-stratified graph \( G \) and a color \( X \) of \( G \), the problem of finding a minimum order tree that spans all the \( X \)-colored vertices is NP-hard.

In our quest to find optimal subgraphs of a \( k \)-stratified graph, we encounter the problem of finding a cycle of minimum order in \( G \) that contains all the \( X \)-colored vertices in \( G \).

**MINIMUM ORDER CYCLE**

**INSTANCE:** Given a \( k \)-stratified graph \( G = (V, E) \), a color \( X \) of \( G \) and a positive integer \( M \leq V \).

**QUESTION:** Is there a cycle of order \( p \leq M \) in \( G \) containing all the \( X \)-colored vertices?
**Theorem 31** Finding a cycle of minimum order containing all vertices of a fixed color in a $k$-stratified graphs is NP-hard.

**Proof:** Let $G$ be a graph. Add a vertex $u$ to $G$ and join $u$ to every vertex of $G$. Further, let $\text{color}(u) = \text{blue}$ and $\text{color}(v) = \text{red}$ for all $v \in V(G)$. Then answering this question in $G + K_1$ is equivalent to finding a Hamiltonian cycle in $G$. However, it is known that finding a Hamiltonian cycle is NP-hard. □

### 4.3 Color-Spanning Subgraphs

In this section we are concerned with finding connected subgraphs that are as stratified (or colorful) as possible. A subgraph containing one vertex from each stratum is referred to as a **rainbow subgraph**. A connected subgraph of smallest order that has at least one vertex from each stratum is referred to as a **minimum color-spanning subgraph** because it spans all the colors and has minimum order.

#### 4.3.1 Color-Spanning Cliques

Given a $k$-stratified graph $G$, a **minimum color-spanning clique** $K_c$ of $G$ is a complete minimum color-spanning subgraph of $G$. A **rainbow clique** $K_R$ of $G$ is a complete rainbow subgraph of $G$. We now concern ourselves with the problem of finding a rainbow-clique of a stratified graph.

If $G$ has a rainbow-clique, then indeed there is a $k$-subset $V_k$ of the vertices of $G$ with exactly one vertex from each stratum such that $\langle V_k \rangle$ induces a complete subgraph of $G$. Conversely, if there is a $k$-subset $V_k$ of $V(G)$ such that $V_k$ has exactly one vertex from each stratum of $G$ and $\langle V_k \rangle$ is complete, then, by definition, $\langle V_k \rangle$ is a rainbow-clique of $G$. Also, observe that the most time-consuming step in the $\text{Find\_Rainbow\_Clique}(G,k)$ algorithm is Step 2. Since $n_t < n$, it follows
Algorithm Find_Rainbow_Clique(G, k)

Input: Graph G = (V,E), k.
Output: A rainbow clique subgraph of G.

Find_Rainbow_Clique(G, k)
1. Find the color density vector (n1, n2, ..., nk) of G.
2. Consider the ri × ri × ... × ri possible k-subsets of the vertices.
3. If any of them induces a complete subgraph, then this k-tuple corresponds to a rainbow-clique.
4. Otherwise none exists.

Figure 37. Algorithm Find_Rainbow_Clique.

that n1 × n2 × ... × nk ≤ n^k. Hence, we have the following.

Theorem 32 There is an O(n^k) algorithm for finding a rainbow-clique of a stratified graph.

4.3.2 Color-Spanning Paths

In a k-stratified graph, a minimum color-spanning path P_c of G is a path of G that spans all the colors in the color spectrum of G. A color-spanning path of order k is called a rainbow path. Our main interest in this section is to determine whether G has a rainbow path and how to find such a path, if it exists. We are also interested in finding the existence of a minimum color-spanning path in G and a procedure for finding one.

Now, we describe an algorithm for finding a rainbow subpath of a path P. Assume that the vertices of P are labeled u_1, u_2, ..., u_n. In the following algorithm, S represents a subset of colors. The algorithm operates by sliding...
a window of size $k$ throughout the path and keeping track of the colors visible through this window.

**Algorithm Find_Rainbow_SubPath($P, k$)**

**Input:** Graph (path) $P = (V, E)$, $k$.
**Output:** A rainbow subpath of $P$.

1. $S \leftarrow \emptyset$, $i \leftarrow 1$
2. While $|S| < k$ and $i < n$ do
3. \hspace{0.5cm} $S \leftarrow \emptyset$
4. \hspace{0.5cm} For $j = i$ to $i + k - 1$ do;
   \hspace{0.5cm} \hspace{0.5cm} $S \leftarrow S \cup \{\text{color}(u_j)\}$
5. \hspace{0.5cm} EndFor
6. \hspace{0.5cm} $i \leftarrow i + 1$
7. EndWhile
8. If $|S| = k$ then
9. \hspace{0.5cm} Return $\langle u_{j-k+1}, u_{j-k}, \ldots, u_j \rangle$
10. Else
11. \hspace{0.5cm} Return "No rainbow subpath exists."

Figure 38. Algorithm Find_Rainbow_SubPath.

Observe that the **While** loop in lines 2-8 is executed at most $n - 1$ times. For each iteration of this **While** loop, the **For** loop in lines 4-6 is executed exactly $k$ times. Hence, $|S| \leq k$ and we have the following.

**Theorem 33** Given a $k$-stratified path $P$ of order $n$. The algorithm $\text{Find}_\text{Rainbow}_\text{SubPath}(P, k)$ finds a $k$-stratified subgraph; with order $k$, of $P$ in time $O(kn)$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The algorithm Find_Rainbow_SubPath(P, k) can be modified to find a minimum color-spanning subpath of a path. In this case, we use a variable size window instead. After each iteration, if a color-spanning subpath is not found yet, we enlarge the window size by 1 until all colors are represented at least once in the window. Note that in the worst case, the original path itself is the color-spanning subpath.

Algorithm Find_Min_Color_Spanning_SubPath(P, k)

Input: Graph (path) P = (V, E), V = k.
Output: A minimum order color-spanning subpath of P.

1. \( m \leftarrow k - 1 \), \( S \leftarrow \emptyset \)
2. While \( |S| < k \) and \( m < n \) do
   3. \( i \leftarrow 1 \)
   4. While \( |S| < k \) and \( i + m < n \) do
      5. \( S \leftarrow \emptyset \)
      6. For \( j = i \) to \( i + m \) do
         7. \( S \leftarrow S \cup \{ \text{color}(u_j) \} \)
      8. EndFor
      9. \( i \leftarrow i + 1 \)
   10. EndWhile
11. \( m \leftarrow m + 1 \)
12. EndWhile
13. Return \( \langle u_{j-m}, u_{j-(m+1)}, \ldots, u_j \rangle \)

Figure 39. Algorithm Find_Min_Color_Spanning_SubPath.

Observe that the width of the window \( m \) may grow from \( k \) to at most \( n \). Each vertex \( v \) of \( P \) is visited at most \( m \) times while the window width is \( m \).
Therefore, in the worst case each of the \( n \) vertices is visited \( \sum_m \) times. It is straightforward to verify the following.

**Theorem 34** Given a \( k \)-stratified path \( P \) of order \( n \). The algorithm \( \text{Find\_Min\_Color\_Spanning\_SubPath}(P,k) \) finds a minimum order color-spanning subgraph of \( P \) in time \( O(n^3) \).

Every tree has at most \( n - 1 \) end-vertices, so \( \text{Find\_Rainbow\_SubPath}(P,k) \) can be used to find the existence of a rainbow path if \( G \) is a tree, say \( T \). There are at most \( (n - 1)(n - 2)/2 \) paths in \( T \) that start and end with an end-vertex. Hence, this gives us an algorithm with the worst case time complexity \( O(kn^3) \). Similarly, we could use the algorithm \( \text{Find\_Min\_Color\_Spanning\_SubPath}(P,k) \) for finding the existence of a color-spanning path of minimum order in any tree \( T \). This provides us with an algorithm for this purpose of time complexity \( O(n^5) \).

### 4.3.3 Color-Spanning Trees

Given a \( k \)-stratified graph \( G \), a *minimum color-spanning tree* \( T_c \) of \( G \) is a minimum order acyclic subgraph of \( G \) that contains at least one vertex from each color class of \( G \). A color-spanning tree of order \( k \) is called a *rainbow tree*. In this section we answer questions about the existence of a rainbow tree in a given stratified graph \( G \). We also describe how to find such a subgraph if it exists and also discuss the problem of finding a minimum color-spanning tree of a graph \( G \).

First we consider the problem of finding a minimum color-spanning tree of a \( k \)-stratified graph \( G \). We show that this problem is NP-hard. The formal statement of a decision version of this problem is given next.

**MINIMUM COLOR-SPANNING TREE (MCST)**
INSTANCE: Given a $k$-stratified graph $G = (V, E)$ and a positive integer $M \leq |V| - 1$.

QUESTION: Is there a tree $T$ in $G$ that spans all colors of $G$ and contains no more than $M$ edges?

Theorem 35 The Minimum Color-Spanning Tree Problem (MCST) is NP-complete.

Proof: It is easy to see that MCST $\in$ NP because a nondeterministic algorithm need only guess a subset $R \subseteq V$ with $|R| \leq M + 1$ and verify in polynomial time whether $\langle R \rangle$ is a subtree of $G$ with at least one vertex from each stratum of $G$.

We transform the Steiner Tree Problem ST in graphs to this problem. Let a graph $G = (V, E)$, a subset $R \subseteq V$, and a positive integer $M \leq |v| - 1$ constitute an arbitrary instance of ST. We now construct a $k$-stratified graph $H$. Let $H \cong G$. Then color each vertex of $G$ in $R$ with a unique color from the set $\{X_1, X_2, \ldots, X_{|R|}\}$ of colors, and assign color $X_{|R|+1}$ to each vertex of $G$ in $V - R$. Finally, let $k = |R| + 1$. This transformation can be done in polynomial time. Observe that every color-spanning subtree of $H$ must contain all vertices corresponding to a vertex from $R$ and at least one other vertex. Hence, a color-spanning subtree of $H$ with no more than $M$ edges corresponds directly to a Steiner tree of $G$ with no more than $M$ edges, and vice-versa. □

We now turn our attention to special cases of this problem. First note that if $G$ is a complete graph on $n$ vertices, then we can start with any vertex, record the color of this vertex in a color list, choose an adjacent vertex of color not yet in the color list, and add the color of the new vertex to the list of colors. We repeat this process until a vertex of each color has been chosen. Since each vertex is adjacent to all other vertices, it is always possible to choose a vertex. Thus,
this algorithm successfully identifies a color-spanning tree (actually a star) of $G$.

Next, we consider the case in which $G$ itself is a tree.

Given a $k$-stratified tree $T$, how can we find a minimum color-spanning subtree of $T$? In the remainder of this section we describe a dynamic programming algorithm for solving this problem when the maximum degree of each vertex is at most $k$. First, however, observe that the following is an immediate consequence of the definition of a minimum color-spanning subtree.

**Lemma 9** The leaves of a minimum color-spanning tree $\tau$ have distinct colors. Furthermore, there is no non-leaf in $V(\tau)$ with the same color as a leaf in $\tau$.

The following result is useful for the analysis of the complexity of our algorithm.

**Lemma 10** In a $k$-stratified graph $G$, a minimum color-spanning tree $\tau$ has at most $k - 1$ leaves.

**Proof:** For $k = 2$ and $k = 3$ this is obvious. So assume that $k > 3$. Suppose to the contrary, that there exists a minimum color-spanning subtree $\tau$ in a $k$-stratified graph $G$ that has $k$ leaves. By the previous lemma, all the leaves must have distinct colors. Also no non-leaf can have the same color as a leaf. This implies that $G$ has $k + 1$ colors which produces a contradiction. □

Consider the 3-stratified tree $T$ illustrated in Figure 40. In this tree each vertex has at most two children. For a vertex $v$ we refer to them as the left child $lchild(v)$ and the right child $rchild(v)$ of $v$.

We now describe a method for finding a minimum color-spanning subtree for this example. Our approach is to first find a minimum color-spanning subtree rooted at $v$ for each vertex $v$ of $T$. Finally, among these, choose one with the
minimum number of vertices. For a vertex $v$ of $T$, we refer to the subtree of $T$ rooted at $v$ as the $v$-subtree of $T$. This terminology is adopted from [F.81].

We now describe how to find a minimum color-spanning subtree rooted at $v_0$. Assume that the tree is rooted at $v_0$. Label the vertices in a breadth-first search manner, from left to right, starting at $v_0$. Let $C_i$ denote a subset of the color spectrum $C$ of $T$. Since $C = \{r, g, b\}$, let $C_1 = \{r\}$, $C_2 = \{g\}$, $C_3 = \{b\}$, $C_4 = \{r, g\}$, $C_5 = \{r, b\}$, $C_6 = \{g, b\}$, $C_7 = \{r, g, b\}$. We find a minimum subtree rooted at each vertex with color spectrum $C_i$, for $1 \leq i \leq 2^k - 1$. A minimum subtree rooted at vertex $v_0$ that spans the color spectrum $C$ is the solution that we are seeking. We refer to such a subtree as a minimum color-spanning $v_0$-subtree because this subtree spans the full spectrum of colors in $T$ and has the minimum number of vertices among all such subtrees rooted at $v_0$.

We start at the leaves of the tree and traverse our way down to the root. Thus, the minimum subtree of children vertices are computed before that of their parents. This enables us to use these results in computing the minimum subtrees for the parents and achieve the time savings that is typical of the dynamic
programming algorithms.

Let \( |T(v_0, \{r, g, b\})| \) denote the number of vertices in a minimum subtree rooted at vertex \( v_0 \) with spectrum \( C = C_7 = \{r, g, b\} \). This quantity is computed as follows. First, note that since \( v_0 \) is red we need not look for another red vertex. Thus

\[
|T(v_0, \{r, g, b\})| = \min\{1 + |T(lchild(v_0), \{g, b\})|, \\
1 + |T(lchild(v_0), \{g\})| + |T(rchild(v_0), \{b\})|, \\
1 + |T(lchild(v_0), \{b\})| + |T(rchild(v_0), \{g\})|, \\
1 + |T(rchild(v_0), \{g, b\})|\}.
\]

In this case, \( lchild(v_0) = v_1 \) and \( rchild(v_0) = v_2 \), and they are both red. So

\[
|T(v_1, \{g, b\})| = \min\{1 + |T(lchild(v_1), \{g, b\})|, \\
1 + |T(lchild(v_1), \{g\})| + |T(rchild(v_1), \{b\})|, \\
1 + |T(lchild(v_1), \{b\})| + |T(rchild(v_1), \{g\})|, \\
1 + |T(rchild(v_1), \{g, b\})|\}.
\]

Since \( lchild(v_1) = v_3 \), which is blue, and \( rchild(v_1) = v_4 \) is red, we have

\[
|T(v_3, \{g, b\})| = |T(v_3, \{g\})| = \infty, \\
|T(v_4, \{g, b\})| = |T(v_4, \{g\})| = \infty, \\
|T(v_3, \{b\})| = 1,
\]

and

\[
|T(v_4, \{b\})| = \min\{1 + |T(lchild(v_4), \{b\})|, 1 + |T(rchild(v_4), \{b\})|\} = \min\{1 + |T(v_6, \{b\})|, 1 + |T(v_7, \{b\})|\}
\]
\[ \min\{1+1, 1+1\} = 2. \]

Notice that for leaves (e.g., \( v_3 \)), these values are either 1 or \( \infty \) and are computed in one step. However, for each non-leaf, the number of vertices in a \( C_i \)-spectrum minimum \( u \)-subtree requires finding the minimum among several other quantities. The number of quantities that need to be computed depends on the degree of \( u \) and the cardinality of \( C_i \). For example, to compute \(|T(v_1, \{g, b\})|\), we must know the number of vertices in six other rooted subtrees. Note that in this case, \( v_1 \) has two children and \(|C_6| = \{g, b\} = 2\). However, since \(|T(v_2, \{g, b\})| = 1 + |T(v_5, \{g, b\})|\), we are required to find the number of vertices in the \( v_5 \)-rooted \( \{g, b\} \)-spectrum minimum subtree.

In general, if we assume that the maximum degree of \( T \) is at most \( k \), then this step requires \( t_i \) computations, where \( t_i \leq (2^{k-1})^{k-1} \). Because, as it was proved in Lemma 10, a color-spanning minimum subtree of a \( k \)-stratified graph has at most \( k - 1 \) leaves, and each child gives rise to at most \( 2^{k-1} \) potentially interesting rooted \( C_i \)-spectrum minimum subtrees. Therefore, since the maximum degree of \( T \) is \( k \), and \( k \) is fixed, \( t_i \) is bounded by a constant. An application of this algorithm to the example given in Figure 40 is illustrated in Table 4.3.3.

Since the maximum degree of a tree can be as large as the size of the tree, it follows that the algorithm discussed here is not polynomial in the general case. Next we present a polynomial time algorithm that is independent of the maximum degree of the tree.

Consider the following algorithm: (1) Select a vertex from each color class, (2) Determine the smallest subtree that covers all selected vertices in step (1). Since a minimum color-spanning subtree contains at least one vertex from each color class then such a subtree must be one of the subtrees found by our algorithm.
Observe that the complexity of this algorithm depends solely on the number of color classes (strata) and the number of vertices in each color class (stratum). Since the number of vertices in each color class is bounded by the order of the tree $n$, and the number of color classes is $k$, it follows that, the worst case complexity of this algorithm is in the order of $O(n^k)$.

Table 4

The Table Computed by MinColorSpSubtree for a 3-stratified Tree

<table>
<thead>
<tr>
<th>$T(v_i, C_j)$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
<th>$C_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_9$</td>
<td>$\infty$</td>
<td>1</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$v_8$</td>
<td>1</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$v_7$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>1</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$v_6$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>1</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$v_5$</td>
<td>2</td>
<td>1</td>
<td>$\infty$</td>
<td>2</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$v_4$</td>
<td>1</td>
<td>$\infty$</td>
<td>2</td>
<td>$\infty$</td>
<td>2</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$v_3$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>1</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$v_2$</td>
<td>1</td>
<td>2</td>
<td>$\infty$</td>
<td>2</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$v_1$</td>
<td>1</td>
<td>$\infty$</td>
<td>2</td>
<td>$\infty$</td>
<td>2</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$v_0$</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

In the following algorithm, $Children(v_i)$ refers to the set of vertices whose parent is $v_i$. Let $T(v_j, C_i)$ denote the tree rooted at $v_j$ that spans the color spectrum $C_i$, where $C_i \subseteq C$ with $1 \leq i \leq 2^k - 1$. **FIND** converts $T$ into the rooted tree at $v_i$, that is, it labels vertices so that the root is at height 1, its children are at height 2, and so on. **INITIALIZE** initializes the values of the Table matrix to infinity. **FindMinimum** returns the cost of the minimum subtree of $T(u_i)$ spanning all colors of $C_j - Color(u_i)$. To find this value, we consider all possible subsets of each color set $C_j$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
**Algorithm MinColorSpSubtree(T,C)**

**Input:** Graph $T = (V,E)$, $C = \{X_1, X_2, \ldots, X_k\}$.  
It is assumed that the maximum degree of a vertex of $T$ is $k$.  
**Output:** Cost of a minimum ordercolor-spanning subtree of $T$.

```
MinColorSpSubtree(T, C)
1. Let $m \leftarrow 2^k - 1$.
2. Let $MinCost \leftarrow \infty$.
3. For each vertex $v_i \in V(T)$ do
   4. FIND $T(v_i, C)$.
5. INITIALIZE $Table[1..n, 1..m] \leftarrow \infty$.
6. $TotalHeight \leftarrow \text{height}(T(v_i, C))$.
7. For $h \leftarrow \text{height}(T(v_i, C))$ downto 1 do
   8. For each vertex $u_i$ at level $h$ do
      9. If $h = TotalHeight$ then
         10. For $j \leftarrow 1$ to $m$ do
             11. If $C_j = \{\text{Color}(u_i)\}$ then $Table[u_i, C_j] \leftarrow 1$.
         12. EndFor
      13. else For $j \leftarrow 1$ to $m$ do
         14. $Table[u_i, C_j] \leftarrow \text{FindMinimum}(\text{Children}(u_i), C_j) + 1$.
      15. EndFor
   16. EndFor
17. If $Table[v_i, C] \leq MinCost$ then $MinCost \leftarrow Table[v_i, C]$.
18. Return $MinCost$.
```

Figure 41. Algorithm MinColorSpSubtree.
4.4 Domination and Independent Sets

Let $G$ be a $k$-stratified graph with stratification $S = \{S_1, S_2, \ldots, S_k\}$. Occasionally, a transitive relation $R$ is defined on $S$, where if $S_i R S_j$, then we say that the set $S_i$ dominates the set $S_j$. Thus, a stratified graph $G$ can be considered an ordered triple $(S, E, R)$, where $E$ is the edge set, or $G$ is the ordered pair $(S, E)$ if no relation is prescribed or if $R = \emptyset$. A 3-stratified graph $G = (S, E, R)$ is shown in Figure 42, where $S = \{S_1, S_2, S_3\}$ with $S_1 = \{s, t\}$, $S_2 = \{u, v\}$, and $S_3 = \{x, y, z\}$, and where the relation $R$ is described by the digraph $D$.

![Figure 42. A 3-stratified Graph.](image)

4.5 Domination in 2-Stratified Trees

In this section we consider 2-stratified trees with stratification $S = \{S_1, S_2\}$ whose domination digraph is given in Figure 43. As usual, we refer to a vertex of $S_1$

![Figure 43. Domination in a 2-stratified Graph.](image)
as red vertex and a vertex of \( S_2 \) as a blue vertex. Consequently, a dominating set of a tree \( T \) is a set of red vertices that dominate all vertices of \( T \). The domination number of \( T \) is then the minimum number of (red) vertices that dominate all vertices of \( T \).

It is useful to define a few preliminary terms. A red edge (blue edge) is an edge that joins two red (blue) vertices. A red star (blue star) is a star whose central vertex is incident only with blue edges. Thus a blue isolated vertex is a blue star as well. Since a graph has a dominating set if and only if every blue vertex is adjacent to some red vertex, we have the following result.

**Lemma 11** A tree has a dominating set if and only if it has no blue star.

A penultimate vertex is a cut-vertex that is adjacent to a leaf. Since blue edges play no role in the determination of dominating sets, we may assume that the given forest \( F \) contains no blue edges.

Observe that if \( D \) is any dominating set of \( F \), then

1. every isolated vertex of \( F \) must belong to \( D \),
2. some red vertex in a component of order 2 in \( F \) must belong to \( D \),
3. every red leaf of a blue penultimate vertex must belong to \( D \),
4. without loss of generality, we may assume that every red penultimate vertex belongs to \( D \), and
5. every neighbor of a vertex in \( D \) that is not in \( D \) need not belong to \( D \).

We now describe an algorithm for finding a minimum dominating set of a forest \( F \) containing no blue edges. In this algorithm we denote the set of red
penultimate vertices of $F$ by $RP$. Also, $NRP$ denotes the union of the closed neighborhoods of the vertices of $RP$. Furthermore, let $RL$ denote the set of red leaves of blue penultimate vertices of $F$ and let $NRL$ denote the union of the closed neighborhoods of the vertices of $RL$. Also, let $S_1$ denote the set of (red) isolated vertices of $F$.

**Algorithm Find.Min.Dominating($F$)**

**Input:** A 2-stratified forest $F = (V, E)$, Such that $F$ contains no blue star.

**Output:** A minimum dominating set $D$ of $F$.

1. $D \leftarrow D \cup RP$
2. $F \leftarrow F - NRP$
3. $D \leftarrow D \cup RL$
4. $F \leftarrow F - NRL$
5. If $F$ has a component of order 3 or more, then return to Step 2.
6. $D \leftarrow D \cup S_1$
7. $D \leftarrow D \cup S_2$
8. Return $D$, $|D| = \gamma(F)$

Figure 44. Algorithm Find.Min.Dominating($F$).

**4.6 Independence in Stratified Graphs**

In this section our major interest will be in layer assignment problems. First, we consider the following.

**The Simple Layer Assignment Problem in k Layers**

Let $G$ be a $k$-stratified graph with strata $S_1, S_2, \ldots, S_k$. Construct a parti-
tion of $V(G)$ into $k$ independent subsets $V_1, V_2, \ldots, V_k$ such that $V_j \subseteq \bigcup_{i=j}^{k} S_i$ for every integer $j$ ($1 \leq j \leq k$).

We denote the Simple Layer Assignment Problem in $k$ layers by LAP-$k$. In general, LAP-$k$ is a difficult problem. In fact, it is straightforward to show the following.

**Theorem 36** For every integer $k \geq 2$, LAP-$k$ is NP-hard.

We now consider LAP-2. It is convenient to denote the strata here by $S_1$ and $S_{12}$. The problem then is to partition the vertex set of a 2-stratified graph $G$ into independent sets $V_1$ and $V_2$ such that $V_1 \subseteq S_1 \cup S_{12}$ and $V_2 \subseteq S_{12}$. Thus, a necessary condition for LAP-2 to have a solution is that the stratum $S_1$ be independent in $G$ (indeed, $S_1 \subseteq V_1$) and that $S_{12}$ induce a bipartite subgraph of $G$, one of whose partite sets must be $V_2$. Of course, $G$ itself must be bipartite. These conditions, though necessary, are not sufficient, however, for LAP-2 to have a solution. For example, the 2-stratified bipartite graph $G$ of Figure 45 satisfies all these conditions, but $V(G)$ cannot be partitioned into subsets $V_1$ and $V_2$ possessing the desired properties since if $S_1 \subseteq V_1$ and $V_1$ is independent, then $V_2 = S_{12}$, which is impossible since $S_{12}$ is not independent.

![Figure 45. A 2-stratified Graph Satisfying Necessary Conditions for LAP-2.](image)

In order to characterize those 2-stratified graphs for which LAP-2 has a solution, we introduce a new term. Let $G$ be a 2-stratified graph with strata $S_1$ and $S_{12}$. A path $P : v_1, v_2, \ldots, v_m (m \geq 3)$ is called a $(1, 1)$ path if $v_1, v_m \in S_1$ and $v_i \in S_{12}$ for $2 \leq i \leq m - 1$. The path $P$ is called even or odd according to whether
its length \( m - 1 \) is even or odd.

**Theorem 37** Let \( G \) be a 2-stratified graph. Then LAP-2 has a solution if and only if \( G \) is a bipartite graph that contains no odd \((1, 1)\) path.

**Proof:** Let the strata of \( G \) be denoted by \( S_1 \) and \( S_{12} \). Assume first that LAP-2 has a solution \( V_1, V_2 \). Then we have seen that \( G \) is bipartite. Suppose, to the contrary, that \( G \) contains an odd \((1, 1)\) path \( v_1, v_2, \ldots, v_m (m \geq 2) \). Then \( m \) is even; \( v_1, v_m \in S_1 \); and \( v_i \in S_{12} \) for \( 2 \leq i \leq m - 1 \). Consequently, \( v_1, v_m \in V_1 \). Since \( v_1 v_2 \in E(G) \), it follows that \( v_2 \in V_2 \). In general, \( v_i \in V_2 \) if \( i \) is even and \( 1 \leq i \leq m \), which gives \( v_m \in V_2 \), producing a contradiction.

For the converse, assume that \( G \) is a bipartite graph that contains no odd \((1, 1)\) path. We may assume, without loss of generality, that \( G \) is connected. Let \( v \in S_1 \). Further, let \( W_1 \) and \( W_2 \) denote the partite sets of \( G \), labeled in such a way that \( v \in W_1 \).

We claim that every vertex of \( S_1 \) belongs to \( W_1 \). Suppose that this is not the case. Then there are vertices of \( S_1 \) that belong to \( W_2 \). Among the vertices of \( S_1 \), let \( x \) and \( y \) be vertices such that \( x \in W_1, y \in W_2 \), and \( d(x, y) \) is minimum. Let \( P : x = x_1, x_2, \ldots, x_k = y \) be an \( x - y \) path of length \( k - 1 = d(x, y) \). Clearly, \( k - 1 \) is odd. No vertex \( x_i (2 \leq i \leq k - 1) \) belongs to \( S_1 \), for whether \( x_i \in W_2 \) or \( x_i \in W_1 \), the pair \( \{x_i, y\} \), respectively, contradicts the defining property of the pair \( \{x, y\} \). Thus \( x_i \in S_{12} \) for all \( i (2 \leq i \leq k - 1) \). However, then, \( P \) is an odd \((1, 1)\) path, contrary to hypothesis. Thus, \( S_1 \subseteq W_1 \) and LAP-2 has a solution \( V_i = W_i (i = 1, 2) \). □

We now consider another layer assignment problem.

**The Pair Layer Assignment Problem in 2k-1 Layers**

Let \( G \) be a \((2k - 1)\)-stratified graph with strata \( S_1, S_{12}, S_{23}, S_3, \ldots, S_{k-1}, \ldots, S_{2k-1} \),
Construct a partition of $V(G)$ into $k$ independent subsets $V_1, V_2, \ldots, V_k$ such that $V_1 \subseteq S_1 \cup S_{12}, V_k \subseteq S_{k-1} \cup S_k$, and $V_i \subseteq S_{i-1} \cup S_i, S_{i,i+1}$ for $2 \leq i \leq k-1$.

We denote the Pair Layer Assignment Problem in $2k - 1$ Layers by PLAP-$(2k - 1)$.

**Theorem 38** Let $G$ be a 3-stratified graph. Then PLAP-3 has a solution if and only if $G$ is a bipartite graph that contains no $(i, i)$ path for $i = 1, 2$.

**Proof:** Let $G$ be a 3-stratified graph with strata $S_1, S_{12}$ and $S_2$. Assume first that PLAP-2 has a solution in $G$. Then there exists a partition $\{V_1, V_2\}$ of $V(G)$ into independent sets such that $V_1 \subseteq S_1 \cup S_{12}$ and $V_2 \subseteq S_{12} \cup S_2$. Necessarily, then, $G$ is bipartite. Also, $S_1 \subseteq V_1$ and $S_2 \subseteq V_2$ so that every $(i, i)$ path is even.

For the converse, assume that $G$ is bipartite graph containing no odd $(i, i)$ path for $i = 1, 2$. Assume, without loss of generality, that $G$ is connected. We show that PLAP-2 has a solution. Certainly this is the case if $S_1 = S_2 = \emptyset$, so assume that $S_1$, say, is nonempty and let $v \in S_1$. Further, let $W_1$ and $W_2$ denote the partite sets of $G$, so labeled that $v \in W_1$.

We claim that every vertex of $S_1$ belongs to $W_2$. Among the vertices of $S_1$, let $x$ and $y$ be vertices such that $x \in W_1, y \in W_2$, and $d(x, y)$ is minimum. Let $P : x = x_1, x_2, \ldots, x_k = y$ be an $x - y$ path of length $k - 1 = d(x, y)$. Clearly, $k - 1$ is odd. No vertex $x_i$ $(2 \leq i \leq k - 1)$ belongs to $S_1$, for whether $x_i \in W_2$ or $x_i \in W_1$, the pair $\{x, x_i\}$ or $\{x_i, y\}$, respectively, contradicts the defining property of the pair $\{x, y\}$. Thus, $x_i \not\in S_1$ for all $i$ $(2 \leq i \leq k - 1)$. However, then, $P$ is a $(1, 1)$ path, contrary to hypothesis. Thus $S_1 \subseteq W_1$, and, similarly, $S_2 \subseteq W_2$. Therefore, $V_i = W_i$ for $i = 1, 2$ is a PLAP-2 solution in $G$. $\square$
CHAPTER V

APPROXIMATION ALGORITHMS WITH APPLICATIONS

5.1 Introduction

In this section we describe a problem that emenates from the physical
design stage of a VLSI chip design. This is one of the problems that motivated
our study of stratified graphs. This problem is concerned with routing of wires
in a multilayer VLSI chip and deals with net selection for over-the-cell (OTC)
routing in standard cell design technology. For a detailed explanation of the
terminology and concepts used in this chapter, we refer the reader to [N.A93].
The following brief description is adapted from [MT89]. A circuit consists of
a set of modules(cells), each being a collection of active devices. Standard cell
architecture requires the layout to consist of rectangular cells of the same height.
A net is a subset of points on the boundary of the modules, called terminals. A
circuit layout involves placement, i.e., finding a placement for the modules in the
plane, and routing, i.e., interconnecting the terminals as specified by the nets.
The region of the plane external to the modules, called the routing region, is
partitioned into simpler subregions, called channels. In its most general form a
channel is a rectangle. For each net we determine a sequence of channels through
which the wires connecting terminals in the net must pass. This is called the
channel routing problem. In [G.85] it was shown that the channel routing problem
is NP-complete. As expected, much research has been done in the design of
heuristic algorithms for solving this problem [P.90, JBL90, JL88, E.83, YY87].
In fact, many channel routers can produce solutions that are nearly optimal.
Consequently, as more layers have become available for routing, researchers have focused on the utilization of the area over the cell for this purpose [HSM93]. This problem is known as the over-the-cell routing problem. It follows from the interactibility of the channel routing problem that this problem also has the NP-complete property. Hence, the research focus is again turned to the development of heuristic and approximation algorithms.

The key idea in all existing heuristics [P.90, JBL90, JL88, E.83, YY87, JMN93, NNM91, HSM93] for over-the-cell channel routing is to select a subset of nets that are most suitable for routing over the cells. In [HSM93] the notion of different net types is introduced. It is shown that the classification of nets into three categories, together with the large number of vacant terminals present in most standard cell designs, can be exploited to achieve significant reduction in the channel height and the total number of vias per routing.

In this chapter we review the net classification presented in [HSM93]. We develop a systematic theory for the development of approximation algorithms addressing these particular routing problems. Specifically, we develop an approximation algorithm for two and three layer over-the-cell routing problems. We show that these algorithms provide reasonable approximations to the optimal solution. Finally, experimental results in application of these approximation algorithms to several standard benchmarks, including PRIMARY1 and Deutsch's difficult example, are presented.

The remainder of this chapter is organized as follows: Section 5.2 is a description of the physical model used by our approximation algorithms plus an overview of the OTC algorithm WISER presented in [HSM93]. Section 5.4 contains an overview of our algorithm for the two layer process. In Section 5.5, we discuss the approximation algorithm for the three layer process. Sections 5.6 and
5.7 contain the experimental results and concluding remarks, respectively.

5.2 Cell Model and Preliminaries

In this section, a cell model used for over-the-cell channel routing is presented, along with an overview of the OTC routing algorithms presented in [HSM93].

We use the same physical model of standard cell, as reported in previous research [JL88, HSM93]. In two layer and three layer standard cell designs, the first metal layer (M1) is reserved for internal cell wiring and feedthrough routing. Therefore, only the second metal layer (M2) and the third metal layer (M3) (if available) are used for routing over the cells. Power and ground rails are routed in the center of the area over the cell rows. This partitions the M2 layer horizontally leaving two rectangular over-the-cell routing regions, one bordering on each terminal row of the cells. The number of tracks available for over-the-cell routing in each rectangular region is determined by the height of the standard cells and may vary depending on the cell library used. Although our result applies to all cell heights, we assume, for the sake of explanation, a 108\(\lambda\) cell height, leaving six tracks available in each rectangle. In the three metal layer model, the entire M3 layer may be used for over-the-cell routing; however, in order to avoid routing over power and ground lines, we partition the M3 area horizontally into two rectangular over-the-cell routing regions. As in the M2 layer, each rectangle has six tracks available for routing. The cell terminals appear at the boundaries of the standard cells in the second metal layer and in some technologies vias are prohibited over transistors. For this reason, nets routed in each of the rectangular, over-the-cell routing regions must be non-overlapping. The physical routing models for the two metal layer case is shown in Figure 46.
5.2.1 OTC Algorithm Overview

In this section, we present an overview of the OTC algorithm (WISER) presented in [HSM93]. There are two key ideas in this approach; namely, (1) the use of vacant terminals to increase the number of nets which can be routed over the cells, and (2) near optimal selection of most suitable nets for over the cell routing.

We now give an informal description of each of the six steps of algorithm WISER; for details see [HSM93].

(I) Net Classification: Each net is classified as one of three types which, intuitively, indicates the difficulty involved in routing this net over the cells.

(II) Vacant Terminal/Abutment Assignment: Vacant terminals/abutments are assigned to each net depending on its type and weight. The weight of a net intuitively indicates the improvement in channel congestion possible if this net can be routed over the cells.

(III) Net Selection: Among all nets that are suitable for routing over the cells,
we select a maximum weighted subset, which can be routed in a single layer.

(IV) **Over-the-Cell Routing:** The selected nets are assigned exact geometric routes in the area over the cells.

(V) **Channel Segment Assignment:** For multi-terminal nets, it is possible that some net segments are not routed over the cells, and therefore, must be routed in the channel. In this step, we select the "best" segments for routing in the channel to complete the net connection.

(VI) **Channel Routing:** The segments selected in the previous step are routed in the channel using a greedy channel router.

In this chapter we concentrate only on improving one of these steps, namely the one concerned with the selection of appropriate nets for routing over the cells.

### 5.2.2 Vacant Terminals and Net Classification

Suppose that we are given a set \( N = \{n_1, n_2, n_3, \ldots, n_k\} \) of nets in a channel \( C \) with top row \( R_t \) and bottom row \( R_b \). Let \( n_i = \{t_{i1}, t_{i2}, \ldots, t_{ij}\} \), where \( t_{ij} \) is a terminal of the net \( n_i \). A terminal is called *vacant* if it is not used for interconnection by any net. Let the function \( OPP(t_i) \) return the terminal directly across the channel in the same column, let the function \( ROW(t_i) \) return the row to which the terminal \( t_i \) belongs, (either \( R_t \) or \( R_b \)) and let \( COL(t_i) \) return the column of the terminal \( t_i \).

A *vacant abutment* is a column with both terminals vacant. In a typical standard cell design, approximately 50 - 80% of the terminals are vacant and 30 - 70% of the columns are vacant abutments.

We decompose all \( m \)-terminal nets into exactly \( m - 1 \) two-terminal nets at
adjacent terminal locations. For example, a six-terminal net $n_i = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ is decomposed into five two-terminal nets, namely, $n_{i1} = \{t_1, t_2\}$, $n_{i2} = \{t_2, t_3\}$, $n_{i3} = \{t_3, t_4\}$, $n_{i4} = \{t_4, t_5\}$, and $n_{i5} = \{t_5, t_6\}$. Clearly, $m - 1$ two-terminal nets are sufficient to preserve the connectivity of the original $m$-terminal net.

After decomposition, we classify each net as one of three basic types. Let $n_j = \{t_1, t_2\}$ be a net, where $t_1$ (respectively $t_2$) is the leftmost (rightmost) terminal of $n_j$. A net $n_j$ is a Type I(a) net if:

(I) $ROW(t_1) = ROW(t_2) = top$

(II) $OPP(t_1)$ and $OPP(t_2)$ are not both vacant.

or

(I) $ROW(t_1) \neq ROW(t_2)$

(II) $OPP(t_1)$ or $OPP(t_2)$ is vacant and on TOP.

A net $n_j$ is a Type I(b) net if:

(I) $ROW(t_1) = ROW(t_2) = bottom$

(II) $OPP(t_1)$ and $OPP(t_2)$ are not both vacant.

or

(I) $ROW(t_1) \neq ROW(t_2)$

(II) $OPP(t_1)$ or $OPP(t_2)$ is vacant and on BOTTOM.

Type I nets may be routed over the cell rows on either the top or the bottom of the channel. A net $n_j$ is a Type II net if both $OPP(t_1)$ and $OPP(t_2)$ are vacant. Type II nets may be routed over the cells on either side (top or
bottom) of the channel. The net $n_j$ is a Type III net if $ROW(t_1) \neq ROW(t_2)$ and there exists a terminal $t_i$ such that $COL(t_1) < COL(t_i) < COL(t_2)$ subject to the condition that $t_i$ and $OPP(t_i)$ are vacant and $OPP(t_1)$ and $OPP(t_2)$ are not vacant.

The three basic net types are illustrated in Figure 47 along with possible over-the-cell routes for each type. In the remaining sections of this chapter, due to the geometric nature of the discussion, a net $n_i$ will be denoted by its left terminal($l_i$) and right terminal($r_i$) such that $COL(l_i) \leq COL(r_i)$.

A typical channel of a standard cell design contains about 44% type I nets, 41% type II nets, and 10% type III nets.

![Figure 47. Three Basic Net Types.](image)

5.3 Existing Algorithms for the Net Selection

The net selection problem can be stated as follows: Given a set $\mathcal{N}$ of nets, select a maximum-weight subset of nets $\mathcal{N}' \subseteq \mathcal{N}$ such that the nets in $\mathcal{N}'$ can be routed in the area over the cell rows in planar fashion. Algorithm WISER [HSM93]
uses a graph theoretic approach to net selection. An overlap graph $G_O$ is defined for intervals of nets in the set $\mathcal{N}$. The net selection problem reduces to the problem of finding a maximum-weight bipartite subgraph in the overlap graph $G_O$. However, the density of the nets in each partite set must be bounded by a constant $k$, which is the number of tracks available in the over-the-cell region. The problem of finding a bipartite subgraph of maximum order in an overlap (circle) graph is known to be NP-complete [MT89], and an 0.75 approximation algorithm is developed.

It is easy to see that there are several restrictions on assignment of vertices to partite sets. For example, a vertex corresponding to a Type I(a) net may not be assigned to a partite set which is to be routed over the lower row of cells. On the other hand, a vertex corresponding to a Type I(b) net may be assigned to the partite set which is to be routed in the lower OTC area. As noted earlier, vertices representing Type II nets may be assigned to either partite set since these nets can be routed over either the upper or lower cell row. A Type III net is partitioned into two Type I nets at the location of its designated abutment.

Let $V_1$ (respectively, $V_2$) be the set of nets that can be routed in the top (bottom) OTC area and let $V_{12}$ represent the nets which are routable in either the top or bottom area. Note that $V_1$ represents the nets in $\mathcal{N}_{I(a)} \cup \mathcal{N}_{II(a)}$, while $V_2$ represents the nets in $\mathcal{N}_{I(b)} \cup \mathcal{N}_{III(b)}$. The set $V_{12}$ represents nets in $\mathcal{N}_{II}$. Let $V'_1 \subseteq V_1 \cup V_{12}$ be the set of nets that are selected for routing in the top OTC area. Similarly, let $V'_2 \subseteq V_2 \cup V_{12}$ be the set of nets that are selected for routing in the bottom OTC area. Note that $V'_1 \cap V'_2 = \emptyset$. Finally, let $S' = (V'_1, V'_2)$ denote the solution.

The algorithm FIS (Fixed Independent Sets) for approximating a maximum-weight bipartite subgraph in $G_O$, presented in [HSM93] guarantees that the den-
sity of each partite set is at most the maximum number of tracks available over the cell rows, that is, \( k \). Let the weight of a vertex \( v \) be denoted by \( w(v) \). The weight of a set of vertices is the sum of the weights of all the vertices in the set. The maximum weight \( k \)-density bipartite subgraph is approximated by finding two maximum weight independent sets in the graph \( G_O \), one for \( V_1' \) and one for \( V_2' \). Since some nets may be assigned to either partite set, the order in which the partite sets are selected is extremely important. In the FIS approximation, both possible orderings are considered. Note that the procedure Max_Independent_Set finds a maximum-weight independent set in an overlap (circle) graph such that the set can be routed using \( k \) tracks in \( O(kn^2) \) time.

In algorithm FIS, the interest is in solving the maximum-weighted bipartite subgraph problem where the density of the partite sets is restricted to a constant \( k \). It is easy to see that the algorithm FIS does indeed satisfy the density requirement since the maximum independent set algorithm used satisfies that requirement.

5.4 Improved Algorithm for the Two Layer Process

The 75% approximation obtained through the algorithm FIS does not take into account the fact that some nets are only routable in upper or lower OTC areas. To clarify this point and adjust for the deficiency, we introduce some notation first. Let \( S_1^* \subseteq V_1 \), \( S_2^* \subseteq V_2 \), and \( S_{12}^* \subseteq V_{12} \). All terms with star (*) as the superscript refer to the optimal solution, while the primed terms refer to the solution selected by the algorithm.

So \( S^* = S_1^* \cup S_2^* \) and \( S_1^* \cap S_2^* = \emptyset \). Also \( S_{12}^* \subseteq S^* \) and \( S_{12}^* \subseteq V_{12} \). That is, \( S_{12}^* \) consists of those vertices in \( S^* \), the optimal solution, that are selected from \( V_{12} \). In addition, let \( S_{12}^* = S_{12-1}^* \cup S_{12-2}^* \). That is, \( S_{12-1}^* \subseteq V_{12} \), which is also a subset of \( V_1' \). In other words, by the subscript 12-1 we wish to indicate
the portion of the first partite set of the optimal solution $S^*$ that are selected from the set $V_{12}$. Similarly, the primed version of these subsets denotes equivalent subsets derived by the algorithm FIS. For example, $S'_{12-2}$ denotes that portion of the second partite set of the solution that has come from $V_{12}$.

In this section, we present an approximation algorithm which promises a solution that is at least 60% of the optimal solution. During this process, we shall introduce a systematic approach that may be utilized to develop approximation algorithms, which promise guaranteed lower bounds on their worst case performance.

We now present a new algorithm 2OTC, which improves the guaranteed lower bound to 60% of the optimal solution. The basic idea of the algorithm is to use three different strategies for selecting the sets. Each strategy works well only for some range of $\alpha$, where $\alpha$ is the ratio indicating the contribution of $V_{12}$ to the optimal solution. When all three are combined we achieve the desired performance guarantee. The formal algorithm 2OTC is shown in Figure 48. In this algorithm $M2IS.CIRCLE$ stands for the maximum bipartite subgraph algorithm, which is described in [JMN93].

**Theorem 39** Let $S^*$ denote the optimal maximum bipartite special subgraph of an overlap graph, and let $S'$ be the solution obtained by algorithm 2OTC. Then

$$\frac{|S'|}{|S^*|} \geq 0.60$$

is a lower bound on the approximation which can be achieved in $O(n^2)$ time.

**Proof:** The $M2IS.CIRCLE$ approximation algorithm as described in [JMN93] produces a solution that is 75% of the optimal solution for $k = 2$. So if we let
**Algorithm 2OTC()**

**Input:** Overlap Graph $G = (V, E)$, $V_1$, $V_2$, $V_{12}$.

**Output:** Set of vertices representing 2-independent sets.

```
begin
  $V_1[1]'$, $V_2[1]' = MIS\_CIRCLE(V_{12})$;
  $V_1[2]' = MIS\_CIRCLE(V_1 \cup V_{12}$; $Temp = V_1[2] \cap V_{12}$;
  $V_2[2]' = MIS\_CIRCLE(V_2 \cup (V_{12} - Temp))$;
  $V_2[3]' = MIS\_CIRCLE(V_2 \cup V_{12});$ $Temp = V_2[3]' \cap V_{12}$;
  $V_1[3]' = MIS\_CIRCLE(V_1 \cup (V_{12} - Temp))$;
  $S' = \emptyset$;
  for $i = 1$ to $3$ do
    $S' = \text{Largest}(S', V_1[i]' \cup V_2[i]'$);
  return $S'$;
end.
```

Figure 48. Algorithm 2OTC.

$\alpha = |S_{12}'|/|S^*|$, then line 1 of the algorithm 2OTC produces a solution that is $0.75 \cdot \alpha \cdot |S^*|$.

Note the symmetric nature of lines 2 and 3 of the algorithm 2OTC. We now show that either line 2 or line 3 produces a solution that is at least $0.5 \cdot \alpha |S^*|$.

Clearly, $|S_2'| \geq |S_2^*|$. Also, $|S_1' \cup S_{12-1}'| \geq |S_1^* \cup S_{12-1}^*|$. Thus,

$$|MIS(V_1 \cup V_{12})| + |MIS(V_2)| = |S_1' \cup S_{12-1}'| + |S_2'|$$

$$\geq |S_1^* \cup S_{12-1}^*| + |S_2^*| \text{ (since } S_1^* \cap S_{12-1}^* = \emptyset).$$

$$= |S_1^*| + |S_{12-1}^*| + |S_2^*| \geq (1 - \alpha)|S^*| + |S_{12-1}^*|.$$ 

either $|S_{12-1}^*| \geq 0.5 \cdot \alpha |S^*|$ or $|S_{12-2}^*| \geq 0.5 \cdot \alpha |S^*|$.
So, \( (1 - \alpha)|S'| + 0.5\alpha|S'| \geq (1 - 0.5\alpha)|S'|. \)

Therefore, \( |S'| = \max\{0.75\alpha, 1 - 0.5\alpha\} \cdot |S'|. \) It is easy to see that the value of the max function is always at least 0.60 with equality if \( \alpha = 0.8. \)

\[ \]

Figure 49. Performance of Algorithm 2OTC.

5.5 Approximation Algorithm for Three Layer Process

In this section our goal is to find four independent subsets \( S_1', S_2', S_1'', S_2'' \) in \( N \) such that no two nets in a given subset intersect and \( |S_1' \cup S_2' \cup S_1'' \cup S_2''| \) is maximum. This is equivalent to the problem of finding the maximum 4-independent subset of the underlying graph. In this case, \( S_1' \) and \( S_2' \) contain those nets that will be routed on the M2 layer. Also \( S_1'' \) and \( S_2'' \) contain those nets that may be routed on the M3 layer. The algorithm is formally represented in Figure 50. It is easy to establish that this algorithm provides a solution that is at least as large as 0.484 times the size of the optimal solution. See Figure 51.
Algorithm 3OTC

**Input:** Graph $G = (V, E)$, $V_1$, $V_2$, $V_{12}$.
**Output:** Set of vertices representing 4-independent sets.

begin

$V_1[1]', V_1[1]'', V_2[1]', V_2[1]''' = \text{M4IS.CIRCLE}(V_{12})$;
$V_1[2]', V_1[2]''' = \text{M2IS.CIRCLE}(V_1 \cup V_{12})$;
$\text{Temp} = V_1[2]' \cap V_1[2]''' \cap V_{12}$;
$V_2[2]', V_2[2]''' = \text{M2IS.CIRCLE}(V_2 \cup (V_{12} - \text{Temp}))$;
$V_2[3]', V_2[3]''' = \text{M2IS.CIRCLE}(V_2 \cup V_{12})$;
$\text{Temp} = V_2[3]' \cap V_2[3]''' \cap V_{12}$;
$V_1[3]', V_1[3]''' = \text{M2IS.CIRCLE}(V_1 \cup (V_{12} - \text{Temp}))$;
$S' = \emptyset$;
for $i = 1$ to $3$ do

$S' = \text{Largest}(S', V_i[i]' \cup V_i[i]''' \cup V_2[i]' \cup V_2[i]''' )$;
return $S'$;

end.

Figure 50. Algorithm 3OTC.

5.6 Experimental Results

The following tables indicate that in both of the benchmarks, the distribution of nets plays a significant role which cannot be ignored. The WISER algorithm presented in [HSM93] was the first work that considered these criteria, however; the selection of nets based on their type classification was not thoroughly understood. As a result, the worst case lower bounds presented here were not attained. Our generic approach reduces the problem to that of selecting independent sets from different subsets of vertices in an underlying graph model of the problem.
In this chapter we presented two approximation algorithms for over-the-cell routing. In both cases we were able to guarantee a reasonable performance. Both algorithms were based on the study of the possible distribution of the optimal solution. The key to our approach was the classification of the nets that could potentially contribute to the optimal solution. This classification partitioned the vertex set of the corresponding overlap graph into three subsets $V_1$, $V_{i2}$, and $V_2$, thereby resulting in a 3-stratified graph. We presented approximation algorithms for finding maximum 2- and 3-independent subsets of the resulting 3-stratified graph. These algorithms were designed through utilization of the optimal polynomial algorithms for the uncolored (unstratified) versions of the problems. We defined a parameter $\alpha$ which served as an indicator of the distribution of nets in the optimal solution. We then provided algorithms which performed reasonably only in a subset of the range of all possible values for $\alpha$. We then showed how to combine these algorithms to achieve a reasonable performance for all values of $\alpha$. This is a general approach and can be applied successfully to similar problems.

Figure 51. Three Layer Approximation.
### Table 5

**Experimental Results - PRIMARY 1**

<table>
<thead>
<tr>
<th>Channel No.</th>
<th>( V_1 ) Nets</th>
<th>( V_{12} ) Nets</th>
<th>( V_2 ) Nets</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( FIS )</td>
<td>( 2OTC )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>80</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td>36</td>
<td>102</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>63</td>
<td>66</td>
<td>114</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>70</td>
<td>72</td>
<td>109</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>69</td>
<td>73</td>
<td>98</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>102</td>
<td>103</td>
<td>101</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td>76</td>
<td>79</td>
<td>105</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>117</td>
<td>111</td>
<td>83</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>91</td>
<td>96</td>
<td>94</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>71</td>
<td>72</td>
<td>104</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>75</td>
<td>77</td>
<td>72</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>73</td>
<td>77</td>
<td>79</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>58</td>
<td>65</td>
<td>114</td>
<td>9</td>
</tr>
<tr>
<td>14</td>
<td>65</td>
<td>66</td>
<td>104</td>
<td>9</td>
</tr>
<tr>
<td>15</td>
<td>77</td>
<td>76</td>
<td>85</td>
<td>7</td>
</tr>
<tr>
<td>16</td>
<td>57</td>
<td>58</td>
<td>82</td>
<td>7</td>
</tr>
<tr>
<td>17</td>
<td>50</td>
<td>53</td>
<td>112</td>
<td>9</td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td>5</td>
<td>44</td>
<td>4</td>
</tr>
</tbody>
</table>

### Table 6

**Deutsch Difficult Example**

<table>
<thead>
<tr>
<th>Channel No.</th>
<th>( V_1 ) Nets</th>
<th>( V_{12} ) Nets</th>
<th>( V_2 ) Nets</th>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( FIS )</td>
<td>( 2OTC )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>30</td>
<td>91</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
CHAPTER VI

OPEN PROBLEMS

We conclude this dissertation by briefly reviewing some of the questions raised earlier in the dissertation. We also introduce a few concepts and questions not discussed previously.

**Question 1** Given a degree-vector multisets of a stratified graph $G$, for each vertex $v$ of $G$, is it possible to determine the number $d_u(v)$ of unicolored edges and the number $d_b(v)$ of bicolored-edges incident to $v$? Is it possible to determine the total number $D_u$ of unicolored edges and the number $D_b$ of bicolored edges in $G$? Are these problems NP-complete?

**Question 2** How many $k$-stratified graphs are there on $n$ vertices for a fixed $k$ ($1 < k < n$)?

**Question 3** Given a finite group $\Gamma'$ and a subgroup $\Gamma$ of $\Gamma'$, does there exist a stratified graph $G$ such that $A(G) \cong \Gamma$ and $A(G') \cong \Gamma'$, where, recall, $G'$ is the underlying graph of $G$?

The similarity of the concept of color patterns to that of degree in ordinary graphs suggests the following question:

**Question 4** Characterize these sets of color patterns that are graphical.

**Question 5** Is the GRAPHICABILITY problem NP-complete?

**Question 6** Characterize the $k$-vector multisets that are degree-vector multisets of other special subclasses of graphs.
Question 7 Investigate the impact of stratifications, such as proper coloring, on the degree-vector multisets, $X$-eccentricity, $X$-center, $X$-periphery, and other graph parameters defined in this dissertation. We studied the impact of stratification on graph parameters briefly in the context of color-regular graphs in Chapter 2 and in the context of the impact of proper coloring on the center of the tree in Section 3.4. Other variants of coloring discussed in the literature such as defective coloring, list coloring, and $T$-coloring suggest possible apriori rules that can be used for stratification.

A vertex $u$ in a $k$-stratified tree $T$ is a color-cut vertex if none of the components of $T - u$ have the same color-spectra number as $T$ or, in other words, if the maximum of the color-spectra number of the components of $T - u$ is strictly less than the color-spectra number of $T$, where the maximum is taken over all components of $T - u$

Definition 1 (Alternate definition) A vertex $u$ in a $k$-stratified tree $T$ is a color-cut vertex if there exists a component of $T - u$ with color-spectra number strictly less than the color-spectra number of $T$.

Note that any vertex $u$ that is a color-cut vertex according to the first definition is also a color-cut vertex according to the alternate definition. Thus, the alternate definition gives rise to more color-cut vertices than the first one. Also, note that both definitions can be generalized to graphs. The intuitive idea behind both definitions is that these vertices somehow partition the colors. However, in our opinion the first definition captures this idea better than the second. That is, according to the first definition, if a vertex $v$ is a color-cut vertex in $T$ and is removed, then there is no subset of colors in any component of $T - v$ that
spans all the colors present in the original tree. However, if we use the alternate definition, then there might still exist subsets of colors, representing the colors of a components of $T - v$ with all the original colors present.

**Question 8** *Does there exist a tree with no color-cut vertex?*

This question has an affirmative answer. If the colors of a tree are well distributed among its vertices, then the tree can have no color-cut vertices. Many stratified trees don't have color-cut vertices, however. For example, Figure 52 shows a 4-stratified tree with no color-cut vertex.

![Figure 52. A 4-stratified Tree With No Color-cut Vertex.](image)

**Question 9** *Does there exist a tree such that all vertices are color-cut vertices?*

Certainly, any $k$-stratified tree with $k$ vertices has this property.

Next, we describe a simple algorithm for finding the color-cut vertices of $G$. 

---

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
For each vertex $v \in V(T)$,

1. Remove $v$.
2. Find $\kappa = \text{the maximum number of colors}$
   in each component of $T - v$.
3. If $\kappa < k$, then $v$ is a color-cut vertex.

Step 2 requires $O(n)$, so the complexity of this algorithm is $O(n^2)$.

Given a $k$-stratified graph $G$ from a family $F$ of $k$-stratified graphs, what is the maximum number of color-cut vertices that $G$ can have?

Clearly, we can have $k$ color-cut vertices if $G = P_k$, i.e., if $G$ is the $k$-stratified path of order $k$.

Can we have more than $k$ color-cut vertices?

**Question 10** If $T$ is a $k$-stratified tree with no color-cut vertices, what can we say about such a tree?

**Lemma 12** Every color-spanning subtree of a tree $T$ contains all color-cut vertices of $T$

**Proof:** We proceed by contradiction. In a $k$-stratified tree $T$, let $w$ be a color-cut vertex of $T$. Assume that $\tau$ is a color-spanning subtree of $T$ such that $w \notin V(\tau)$. Since $\tau$ is a color-spanning subtree of $T$, for all $u, v \in V(\tau)$ there is a $u - v$ path in $\tau$ that does not go through $w$. This implies that $\tau$ is a subtree of a connected component of $T - w$, say $T'$. Thus, number of colors in $T'$ is the same as the number of colors in $T$. But this contradicts the fact that $w$ is a color-cut vertex since, by definition of color-cut vertex, the color-spectra number of all connected components of $T - w$ must be strictly less than that of $T$. □
The next example shows a minimum-color spanning tree. Note that in this tree all the end-vertices are unique with respect to their color. Also note that the center of this tree has a total proximity much larger than the minimum total proximity in $T$. See Figure 53.

![Graph T](image)

Figure 53. The Total Proximity of $v$ is Greater Than the Minimum Total Proximity.

Next we examine the relationship between the total proximity of a vertex and a minimum color-spanning subtree. The following example shows that a minimum color spanning subtree may possess vertices with total proximity as large as the largest total proximity in a tree. Note that this minimum color spanning subtree does not contain a vertex of minimum proximity among all the vertices of the original tree. See Figure 54.

However, note that the number of edges in a minimum-color spanning subtree of a tree $T$ is always less than the minimum total proximity.

In a $k$-stratified graph $G$, recall that $\Delta_{X_i}(G)$ denotes the maximum $X_i$-proximity of $G$. This suggests the following question.
Figure 54. The Vertex $v$ With Minimum Total Proximity Does Not Belong to the Minimum Color-spanning Subtree

Question 11 Let $G$ be an ordinary graph with $n$ vertices, and let $\vec{N} = (n_1, n_2, \ldots, n_k)$ denote a $k$-vector of positive integers such that $\sum n_i = n$. What is the maximum $\Delta x_i$ ($1 \leq i \leq k$) among all stratifications of $G$ with color density vector $\vec{N}$? For example, it is obvious that if $G \cong K_n$, then $\Delta x_i = 1$ ($1 \leq i \leq k$) over all stratifications of $G$.

Question 12 Define and study the concept of $X$-median in a $k$-stratified graph.

A $k$-stratified graph $G$ can be rotated into a $k$-stratified graph $H$ if $G$ contains distinct vertices $u$, $v$, and $w$ such that $uv \in E(G)$, $uw \not\in E(G)$, and $H \cong G - uv + uw$. More generally, a $k$-stratified graph $G$ can be $r$-transformed into a $k$-stratified graph $H$ if there exists a sequence $G \cong H_0, H_1, \ldots, H_n \cong H_n$ ($n \geq 0$) of $k$-stratified graph such that for $n \geq 1$, $H_i$ can be rotated into $H_{i+1}$ for $i = 0, 1, \ldots, n-1$. We note that $r$-transformation is an equivalence relation on the set of all graphs. If a $k$-stratified graph $G$ can be $r$-transformed into a $k$-stratified graph $H$, then $G$ and $H$ have the same size and the $i$th strata ($i = 1, 2, \ldots, k$) of $G$ and $H$ have the same cardinality. It is perhaps less clear that the converse of this implication is true as well. The proof of this follows in the same manner as the proof of Proposition 1 in [GFH85].
Theorem 40 A $k$-stratified graph $G$ can be $r$-transformed into a $k$-stratified graph $H$ if and only if $G$ and $H$ have the same size and the corresponding strata have the same cardinality.

Let $G$ and $H$ be $k$-stratified graphs of the same size, where corresponding strata have the same cardinality. Then the distance $d(G, H)$ is the smallest nonnegative integer $n$ for which there exists a sequence $G \cong H_0, H_1, \ldots, H_n \cong H(n \geq 0)$ of $k$-stratified graphs such that for $n \geq 1$, $H_i$ can be rotated into $H_{i+1}$ for $i = 0, 1, \ldots, n - 1$.

A greatest common subgraph of 2-stratified graphs $G_1$ and $G_2$ can be defined. Denote the set of greatest common subgraphs of $G_1$ and $G_2$ by $gcs(G_1, G_2)$.

Question 13 For every positive integer $n$, do there exist 2-stratified graphs $G_n$ and $H_n$ such that $G_n \cong H_n$ and $|gcs(G_n, H_n)| = n$?

Question 14 For every 2-stratified graph $G$ without isolated vertices, do there exist 2-stratified graphs $G_1$ and $G_2$ of equal size such that $gcs(G_1, G_2) = \{G\}$? Assuming that the answer is yes, can we add the extra condition that $G_1 \cong G_2$?

Question 15 Find two 2-stratified graphs $H_1$ and $H_2$ of equal size for which there are no 2-stratified graphs $G_1$ and $G_2$ such that $gcs(G_1, G_2) = \{H_1, H_2\}$.

Question 16 For every pair $m, n$ of integers with $m \geq 2$ and $n \geq 1$, do there exist pairwise nonisomorphic 2-stratified graphs $G_1, G_2, \ldots, G_m$ of equal size such that $|gcs(G_1, G_2, \ldots, G_m)| = n$?

Question 17 For every 2-stratified graph $G$, do there exist pairwise nonisomorphic 2-stratified graphs $G_1, G_2, G_3$ of equal size such that $gcs(G_1, G_2, G_3) = \{G\}$?
BIBLIOGRAPHY


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


