Spring 1997

Spiralling Through Graph Theory

Melinda Milani
Carroll College, Helena, MT

Follow this and additional works at: https://scholars.carroll.edu/mathengcompsci_theses

Part of the Discrete Mathematics and Combinatorics Commons, and the Secondary Education Commons

Recommended Citation

This Thesis is brought to you for free and open access by the Mathematics, Engineering and Computer Science at Carroll Scholars. It has been accepted for inclusion in Mathematics, Engineering and Computer Science Undergraduate Theses by an authorized administrator of Carroll Scholars. For more information, please contact tkratz@carroll.edu.
SPIRALLING THROUGH GRAPH THEORY

An Honors Thesis

Submitted to the Department of Mathematics,
Engineering, Physics, and Computer Science
in Partial Fulfillment of the Requirements to Graduate with Honors.

by MELINDA MILANI

Carroll College
Helena, Montana
April 1997
# TABLE OF CONTENTS

LIST OF ILLUSTRATIONS .............................................................................................................. iv

ACKNOWLEDGEMENTS ................................................................................................................ vi

CHAPTERS:

1. INTRODUCTION
   1.1 Purpose ............................................................................................................................ 1
   1.2 Background of Graph Theory ......................................................................................... 1
   1.3 Preview of Chapters ......................................................................................................... 3

2. GRAPH THEORY
   2.1 Introduction ..................................................................................................................... 4
   2.2 Components of Graphs ................................................................................................. 4
   2.3 Types of Graphs ............................................................................................................. 7
      2.3.1 Simple Graph .......................................................................................................... 8
      2.3.2 Multigraphs ........................................................................................................... 11
      2.3.3 Pseudographs ....................................................................................................... 12
      2.3.4 Directed Graphs ................................................................................................... 13
      2.3.5 Directed Multigraphs ......................................................................................... 14
   2.4 Characteristics of Graphs .............................................................................................. 16
   2.5 Handling Graphs ........................................................................................................... 19
   2.6 Paths and Circuits .......................................................................................................... 21
   2.7 Trees ............................................................................................................................. 25
   2.8 Relationships Between Graphs ..................................................................................... 26
      2.8.1 Duals ...................................................................................................................... 27
      2.8.2 Isomorphisms ....................................................................................................... 30
   2.9 Conclusions .................................................................................................................... 31

3. APPLICATIONS OF GRAPH THEORY
   3.1 Introduction ..................................................................................................................... 32
   3.2 Applications in Chemistry ............................................................................................. 32
   3.3 Applications in Biology ................................................................................................. 34
   3.4 Applications in the Humanities ..................................................................................... 36
      3.4.1 Influence Graphs .................................................................................................. 36
      3.4.2 Sociograms .......................................................................................................... 37
   3.5 Applications in Engineering ......................................................................................... 38
      3.5.1 An Application in Electrical Engineering ............................................................ 38
      3.5.2 An Application in Engineering Mechanics ......................................................... 43
   3.6 Applications in Discrete Mathematics .......................................................................... 48
LIST OF ILLUSTRATIONS

FIGURES:

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>DESCRIPTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>A Labeled Simple Graph</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Graph with an Isolated Vertex</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>Graph with a Bridge and a Loop</td>
<td>7</td>
</tr>
<tr>
<td>2.4</td>
<td>A Complete Graph</td>
<td>8</td>
</tr>
<tr>
<td>2.5</td>
<td>A Cycle</td>
<td>9</td>
</tr>
<tr>
<td>2.6</td>
<td>A Wheel</td>
<td>10</td>
</tr>
<tr>
<td>2.7</td>
<td>A Bipartite Graph</td>
<td>11</td>
</tr>
<tr>
<td>2.8</td>
<td>A Multigraph</td>
<td>12</td>
</tr>
<tr>
<td>2.9</td>
<td>A Pseudograph</td>
<td>13</td>
</tr>
<tr>
<td>2.10</td>
<td>A Digraph</td>
<td>14</td>
</tr>
<tr>
<td>2.11</td>
<td>A Directed Multigraph</td>
<td>15</td>
</tr>
<tr>
<td>2.12</td>
<td>A Graph and its Subgraph</td>
<td>16</td>
</tr>
<tr>
<td>2.13</td>
<td>The Union of Subgraphs</td>
<td>17</td>
</tr>
<tr>
<td>2.14a</td>
<td>A Planar Graph</td>
<td>18</td>
</tr>
<tr>
<td>2.14b</td>
<td>A Plane Graph</td>
<td>18</td>
</tr>
<tr>
<td>2.15</td>
<td>Graph Showing an Eulerian Path</td>
<td>22</td>
</tr>
<tr>
<td>2.16</td>
<td>Graph Showing an Eulerian Circuit</td>
<td>23</td>
</tr>
<tr>
<td>2.17</td>
<td>Graph Showing a Hamiltonian Path</td>
<td>24</td>
</tr>
<tr>
<td>2.18</td>
<td>Graph Showing a Hamiltonian Circuit</td>
<td>25</td>
</tr>
<tr>
<td>2.19</td>
<td>A Tree</td>
<td>26</td>
</tr>
<tr>
<td>2.20</td>
<td>Stage I of Constructing the Dual of a Graph</td>
<td>27</td>
</tr>
<tr>
<td>2.21</td>
<td>Stage II of Constructing the Dual of a Graph</td>
<td>28</td>
</tr>
<tr>
<td>2.22</td>
<td>Stage III of Constructing the Dual of a Graph</td>
<td>29</td>
</tr>
<tr>
<td>2.23</td>
<td>The Dual of the Graph in Figure 2.20</td>
<td>30</td>
</tr>
<tr>
<td>2.24</td>
<td>Isomorphic Graphs</td>
<td>31</td>
</tr>
<tr>
<td>3.1</td>
<td>The Chemical Structure (Graph) of Water</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Chemical Alkanes</td>
<td>34</td>
</tr>
<tr>
<td>3.3</td>
<td>Niche Overlap Graph of a Forest Ecosystem</td>
<td>35</td>
</tr>
<tr>
<td>3.4</td>
<td>An Influence Graph</td>
<td>36</td>
</tr>
<tr>
<td>3.5</td>
<td>A Sociogram</td>
<td>37</td>
</tr>
<tr>
<td>3.6</td>
<td>An Electrical Circuit</td>
<td>39</td>
</tr>
<tr>
<td>3.7</td>
<td>A Spring Assembly</td>
<td>44</td>
</tr>
<tr>
<td>3.8</td>
<td>Forces Acting on Each Joint and the Corresponding Equilibrium Equations</td>
<td>47</td>
</tr>
<tr>
<td>3.9</td>
<td>Bipartite Graph of Possible Marriages</td>
<td>51</td>
</tr>
<tr>
<td>3.10</td>
<td>Bipartite Graph Showing the Paired Couples</td>
<td>54</td>
</tr>
<tr>
<td>3.11</td>
<td>A Computer Network</td>
<td>56</td>
</tr>
<tr>
<td>3.12</td>
<td>Minimum Cost Spanning Tree via Prim's Algorithm</td>
<td>59</td>
</tr>
<tr>
<td>3.13</td>
<td>Minimum Cost Spanning Tree via Kruskals Algorithm</td>
<td>62</td>
</tr>
</tbody>
</table>
Acknowledgements

This thesis did not just spring from my head, entire, as did Athena from Zeus's head. There are many people I wish to acknowledge for assisting me in this sometimes Herculean task. First and foremost, I would like to thank my director Dr. Scharf—the patternologist of patternologists. Dr. Scharf worked with me during the summer previous to my senior year. He met with me weekly until my thesis was finished, but Dr. Scharf didn't just “meet with me.” He provided excellent guidance by asking me questions (even when I didn't want to answer them), answering my questions, and laughing at my math jokes along the way. I also appreciate the use of the department lap top when the school computer lab was unavailable. Dr. Scharf is truly a teacher.

I would also like to acknowledge my readers, Professor Phil Rose and Dr. James Murphy. Professor Rose was a great source regarding graph theory and shared in the excitement of studying this math concept. Dr. Murphy was very supportive every step of the way and provided a welcome counterbalance to my mathematically-minded readers. I especially appreciate the speed with which they returned my drafts for revisions.

The entire math department was very helpful during this process, in some form or another. They helped me most frequently by loaning me and also taking the time to answer “quick” questions that arose along the way. I would especially like to thank Professor Terry Mullen for all of his technical support regarding AutoCad.

Kiely Keane, Kelly Gorin, Allison George, and Kirsten Rindal deserve special acknowledgments for allowing me to vent my frustrations during times of trial and for
supporting me during the entire process. Special thanks to Kiely and Kelly for living with me on a daily basis throughout this process.

Finally I would like to thank my family. They acted like it was natural for me to write a thesis and that this endeavor is definitely within my capabilities. Their unshaking belief was much appreciated, especially when my beliefs were shaking. This thesis is the product of much supported effort—all of these people acknowledged made this final thesis possible because of their own special ways of encouraging me. Thank you to all!
CHAPTER 1

INTRODUCTION

1.1 Purpose

The purpose of this thesis is twofold. The first objective is to explore graph theory—both its concepts and applications. The second objective is to use graph theory in a particular application: education. These two objectives culminate course work done at Carroll College under the program of Mathematics—Secondary Education.

1.2 Background of Graph Theory

It is difficult to choose just one area of mathematics to study because there are many possible areas, each interesting in its own way. Graph theory is the chosen area of study for this thesis because of its discrete mathematics applications and its connection to linear algebra. This is the background of graph theory as it applies to this thesis.

Graph theory is a relatively new area of mathematics, “new” in the sense that graph theory was not studied by ancient Greeks. The seeds of graph theory, however, were present in ancient Greece, but not fully called upon until the eighteenth century.
Even during the 1700s, graph theory was considered more of a game, rather than a branch of mathematics. As Biggs, Lloyd, and Wilson explain,

The origins of graph theory are humble, even frivolous. Whereas many branches of mathematics were motivated by fundamental problems of calculation, motion, and measurement, the problems which lead to the development of graph theory were often little more than puzzles, designed to test the ingenuity rather than to stimulate the imagination [2].

One of the most popular games involved the people of Königsberg and their bridges. The object of the game was to walk through the city, ending where you began your walk and crossing each bridge only once. No one could find a solution, “but it was not until 1736 that the problem was treated from a mathematical point of view and the impossibility of finding such a route was proved” [2]. The mathematician credited with proving the impossibility of solving the Königsberg bridge problem is Leonard Euler.

Euler is considered a titan of mathematics; his prolific work touches many branches of the mathematical tree. More impressive than the sheer number of works Euler contributed is the fact that his contributions are important and are still looked to as pillars of mathematics.

In 1736 when Euler took the Königsberg bridge game seriously, he essentially created graph theory. Euler was so respected in his time that if he took his time to analyze a “game” mathematically, then it must be okay for others to study graph theory as well. It did take some time for graph theory to be considered a serious field of study because not every mathematician enjoyed Euler’s respect in the mathematical community. Currently, graph theory is a very respected branch of mathematics with applications in various fields such as economics, operations research, engineering, biology, sports, and, of course, mathematics. It is ironic to think that graph theory came about because no one could solve the Königsberg bridge problem.
1.3 Preview of Chapters

This thesis is organized in three major parts, separated into chapters. Chapter 2 provides an explanation of graph theory including terminology. Chapter 3 presents a few applications of graph theory. There are many other and more complex applications than can be contained within this thesis. My purpose, however, is to communicate a sense of the power and versatility of graph theory in an understandable manner. Chapter 4 shows how graph theory can be taught in a specific instructional framework known as a spiral curriculum. This chapter concludes by demonstrating this plan via modules that teach students graph theory, at various educational levels. The final chapter, 5, presents my conclusions.
CHAPTER 2

GRAPH THEORY

2.1 Introduction

Graph theory is a close cousin to sets and set theory. Thinking of set theory will provide a frame of reference that will assist in understanding the following terminology. In defining the terms and providing explanations of graph theory I primarily use two sources: Discrete Mathematics and Its Applications by Kenneth H. Rosen and Introduction to Graph Theory by Robin J. Wilson.

2.2 Components of Graphs

Graphs are discrete structures comprised of two components: vertices and edges. A vertex is simply a point in space. Another name for a vertex is a node. The second of the two components of a graph is an edge. Edges connect vertices.

In set notation, a simple graph is denoted elegantly and succinctly by $G = (V,E)$; $G$ is the graph, $V$ is the nonempty set of vertices, and $E$ is the set of ordered pairs of distinct edges. Vertices are named according to the specific application of the graph. For example, vertices can be numbered, lettered, or given the name of a city. Edges are named in a similar fashion to vertices. However, the most common
convention, as indicated by the above definition, is to denote edges with an ordered pair made up of two distinct vertices. Figure 2.1 shows a graph with the vertices labeled $v_n$ and the edges labeled $e_n$.

![Figure 2.1. A Labeled Simple Graph.](image)

Vertices and edges, by definition, are inextricably related. Edges distinguish two characteristics specific to vertices. The first is the characteristic of connectedness. Two vertices are connected when at least one edge is between them. If a vertex is not connected, it is isolated. In Figure 2.2, $v_3$ is an isolated vertex, all of the other vertices are connected.
The second characteristic regarding vertices and edges is degree. The *degree* of a vertex is defined to be the number of edges that connect with it. An isolated vertex, such as $v_3$ in Figure 2.2 has degree 0. A vertex with degree 1 is connected to the graph with a single edge called a *bridge*. In Figure 2.3, edge $e_4$ is the bridge connecting $v_1$ with $v_3$. There is one other special type of edge, called a *loop*. As the name implies, a *loop* leaves and enters the same vertex, as shown on vertex $v_5$ of Figure 2.3.
Thus, an isolated vertex with a loop has a degree of two.

2.3 Types of Graphs

All graphs fall into one of five different classifications. The five types of graphs include simple graphs, multigraphs, pseudographs, directed graphs, and directed multigraphs.
2.3.1 Simple Graphs

I used the definition of a simple graph to introduce vertices and edges. As a reminder, a simple graph is defined as $G = (V,E)$ where $G$ is a graph containing a set $V$, of vertices and $E$, of edges. Each edge connects two distinct vertices; thus no two edges connect the same pair of vertices. Therefore a simple graph cannot have loops. Figure 2.1 is a simple graph.

Within the classification of simple graphs, there are four sub-classes: complete graphs, cycles, wheels, and bipartite graphs. Complete graphs are graphs that have exactly one edge between each pair of distinct vertices. There are no isolated vertices in a complete graph. In Figure 2.4, each vertex is connected to the graph.

Figure 2.4. A Complete Graph.
Cycles are graphs that have the edges ordered in a specific way. Specifically, the edges must cycle through each vertex and end at the beginning vertex. All cycles must have at least three vertices to exist. A cycle with four vertices is defined as: \{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\} and looks like Figure 2.5.

![Figure 2.5. A Cycle.](image)

A wheel or star, as shown in Figure 2.6 is a cycle that has one additional vertex. This new vertex connects to each vertex of the original cycle. The additional vertex gives the graph a radial appearance.
The last subclass of simple graphs are bipartite graphs. These are graphs whose vertices can be partitioned into two nonempty groups, such that every edge connects vertices from opposite groups with the stipulation that no edge can connect two vertices within the same group. As seen in Figure 2.7, $v_1$ and $v_2$ are in one group of vertices, while $v_3$, $v_4$, and $v_5$ are in the other group.
This is bipartite because there is no edge connecting $v_1$ to $v_2$, nor any edges between $v_3$, $v_4$, or $v_5$.

2.3.2 Multigraphs

The mathematical definition of a multigraph is twofold. First, a multigraph is defined, as all graphs are, with $G = (V, E)$. And second, a multigraph is defined with a function $f$ from $E$ to $\left\{ \left\{ u, v \right\} | u, v \in V, u \neq v \right\}$. This means that multiple edges are allowed between vertices. If $f(e_1) = f(e_2)$ then two edges, $e_1$ and $e_2$, are called parallel. In other words, there are two (multiple) edges connecting the two vertices. Loops, however, are not allowed. In Figure 2.8, there are two edges between $v_1$ and $v_4$, $v_3$ and $v_2$, and $v_2$ and $v_4$. 

Figure 2.7. A Bipartite Graph.
2.3.3 Pseudographs

Closely related to a multigraph is a pseudograph. The mathematical definition of a pseudograph is $G = (V,E)$ and a function $f$ from $E$ to $\{\{u, v\}|u, v \in V\}$. The difference between a multigraph and a pseudograph is that loops are allowed in a pseudograph. Figure 2.9 is a pseudograph and has a loop at vertex $v_3$. 
2.3.4 Directed Graphs

The next type of graph in the main classification of graphs is a directed graph, also called a digraph. Directed graphs have the same definition as simple graphs \((G = (V,E))\) with two exceptions. First, the set of edges, \(E\), is comprised of ordered pairs of the elements of \(V\). The first element of the ordered pair denotes the initial vertex and the second element denotes the terminal vertex. Thus, a direction is established for each edge. The second difference between directed graphs and simple graphs is that loops are allowed in directed graphs as seen in Figure 2.10.
2.3.5 Directed Multigraphs

The last type of graph is a directed multigraph. The relationship existing between simple and directed graphs is similar to multigraphs and directed multigraphs. The mathematical definition of a directed multigraph is $G = (V, E)$ and a function $f$ from $E$ to $\{(u, v) \mid u, v \in V\}$. As you can see, $u$ is allowed to equal $v$, which means that not only are multiple edges allowed, but loops are allowed as well. Figure 2.11 has two edges between vertices $v_1$ and $v_2$ and a loop on $v_3$, and thus is a directed multigraph.
Figure 2.11. A Directed Multigraph.

Table 2.1 summarizes the five main classes of graphs.

<table>
<thead>
<tr>
<th>TYPE OF GRAPH</th>
<th>TYPE OF EDGES</th>
<th>MULTI EDGES ALLOWED?</th>
<th>LOOPS ALLOWED?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>Undirected</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Multigraph</td>
<td>Undirected</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Pseudograph</td>
<td>Undirected</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Directed</td>
<td>Directed</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Directed Multigraph</td>
<td>Directed</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 2.1. Classes of Graphs
2.4 Characteristics of Graphs

Graphs can often be complex and unwieldy. Fortunately, when certain applications require only some portions of a complicated graph to be the area of focus, vertices and edges that are not of interest may be removed. What is left after the removal of the extraneous components is a subgraph. Figure 2.12 shows the subgraph of interest, $H = (W,F)$ and the original graph, $G = (V,E)$. By definition of a subgraph, $W \subseteq V$ and $F \subseteq E$.

![Graph and Subgraph](image)

Figure 2.12. A Graph and its Subgraph.
A graph may have several subgraphs which are not necessarily all connected. The union of all connected subgraphs yields a graph. In Figure 2.13, there are two components, H and J, of the graph K. Both H and J are subgraphs of K.

Figure 2.13. The Union of Subgraphs [8].

Besides noting the components of a graph, it is also important to note the plane (or planes) that contain a graph. The term planar is used to describe graphs which can be drawn in one plane without any edges crossing. A crossing is defined to be the intersection of edges at any point other than a vertex. In Figure 2.14a, the edges \( e_5 \) and \( e_6 \) are drawn one on top of the other, but could be drawn in such a way that \( e_5 \) and \( e_6 \)
do not cross. Figure 2.14b connects the vertices the same way as the vertices of Figure 2.14a without any edges crossing.

A graph with no edges crossing, such as Figure 2.14b, is called a plane graph. Contrast plane graphs with non-plane graphs, whose edges in no way can be drawn without crossings.

A plane graph partitions the plane into faces. Each face consists of the set of all points such that any two points in a face, \( x \) and \( y \), can be connected with a Jordan curve (i.e. a continuous arc not crossing any edges) bordered by edges. Faces of this type result in finite areas. For example, inspecting Figure 2.14b gives four faces. Three faces have finite areas, while the fourth has an infinite area. The face having an infinite area is the infinite face. This face is bordered, but not contained, by the edges. Faces are also easily identified with
visual inspection. Trees, which will be introduced later, are graphs whose only face is the infinite face.

2.5 Handling Graphs

Besides naming a graph, we must also look at the components of a graph. One way to look at the components is to simply draw the graph. While this method works because one picture is worth a thousand words, it is often cumbersome and does not lend itself to mathematical operations. In this regard, a better way to represent a graph is to use a matrix, either an incidence matrix or an adjacency matrix.

An incidence matrix describes which edges are incident with each vertex. In other words, for an undirected graph an incidence matrix is an $n \times m$ matrix, $M = [m_{ij}]$. $m_{ij} = 1$ when edge $e_j$ is incident with vertex $v_i$, otherwise $m_{ij} = 0$. If the graph is a directed graph, $m_{ij} = -1$ when edge $e_j$ is initially incident with $v_i$, and $m_{ij} = +1$ when terminally incident with the vertex. Thus incidence matrices apply to graphs with either directed or undirected edges. Matrix 2.1 is the incidence matrix for the digraph in Figure 2.11.
An adjacency matrix describes adjacent vertices. In the language of mathematics, an adjacency matrix is an $n \times n$ matrix $A = [a_{ij}]$, $a_{ij} = 1$ is an edge of the graph, otherwise $a_{ij} = 0$. If the graph is some sort of a multigraph and two edges connect adjacent vertices, then $a_{ij} = n$, where $n$ is the number of edges. An adjacency matrix can also be used to describe digraphs. In this case, $a_{ij} = 1$ if there is an edge from $v_i$ to $v_j$. Matrix 2.2 is the adjacency matrix for Figure 2.11.

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
<th>$v_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$v_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$v_3$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$v_4$</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$v_5$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$v_6$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Matrix 2.2. The Adjacency Matrix for the graph in Figure 2.11.
Both Matrices 2.1 and 2.2 describe the same graph, Figure 2.11. Neither format is better than the other. Choosing which type of matrix to use depends on the context of the application.

2.6 Paths and Circuits

Paths and Circuits fall into two main types. They are either Eulerian or Hamiltonian. A path, in both cases, is no different than the path Little Red Riding Hood took to get to her grandmother’s house. A path is simply a way to get from vertex \( a \) to vertex \( b \). Using the language of mathematics, a path from \( u \) to \( v \) is a sequence of edges \( e_1, \ldots, e_n \) such that \( f(e_1) = \{u, x_1\}, f(e_2) = \{x_1, x_2\}, \ldots, f(e_n) = \{x_{n-1}, v\} \). The initial, \( x_0 = u \) and the terminal vertex \( x_n = v \). The definition of a circuit follows from the definition of a path. The difference is that \( u = v \). In other words, the initial vertex is the same as the terminal vertex. Paths and circuits specially designated as simple maintain sequences where each edge is traversed once.

An Eulerian path is a simple path that contains every edge. In traversing an Eulerian path, a vertex may be used more than once. Figure 2.15 shows a path beginning on vertex \( v_1 \) and ending on vertex \( v_2 \).
Similarly, an *Eulerian circuit* is a simple circuit containing every edge, as Figure 2.16 demonstrates. Vertex $v_1$ is the initial and the terminating vertex of the circuit.
Conversely, *Hamiltonian paths* and circuits focus on vertices, rather than edges. A Hamiltonian path is a path that contains every vertex exactly once. It is not required to traverse every edge. Figure 2.17 shows a Hamiltonian path. The initial vertex of the path is $v_1$ and the terminal vertex is $v_4$. 

**Eulerian Circuit:**

$v_1, v_3, v_5, v_4, v_3, v_2, v_1$. 

Figure 2.16. Graph Showing an Eulerian Circuit.
Similarly, a Hamiltonain path is a *Hamiltonian circuit* if $u = v$. Figure 2.18 shows a Hamiltonian circuit. This circuit begins and ends with vertex $v_1$. 

**Hamiltonian Path:**
$v_1, v_2, v_3, v_4$. 

**Figure 2.17.** Graph Showing a Hamiltonian Path.
2.7 Trees

Trees are a very specific type of graph. Trees that are familiar to most people include family trees and tournament schedules, such as the trees used to determine who plays who, and when, at Wimbeldon. In mathematical terms, a tree is a connected undirected graph with no simple circuits. Figure 2.19 shows a tree. Trees have very important applications as will be described in Chapter 3.
2.8 Relationships Between Graphs

In mathematics, we are often looking for connections or patterns that exist in different sets of data. Some would define mathematics as the study of patterns. Graph theory is no different from any other mathematical study. Two important types of patterns, or relationships, that are examined in certain applications of graph theory are duality and isomorphisms.
2.8.1 Duals

Duality, by definition, implies two different perspectives of one object. For example, the duality of a coin is either heads or tails. Heads and tails represent the two perspectives of the same coin. In graphs, we have three perspectives of a graph to choose from: vertices, edges, and faces. By the definition of duality, however, we are only allowed two perspectives. The number of edges does not change in the construction of a dual. Therefore the dual of a graph examines the relationships between the faces and vertices of a particular graph. This relationship is important because, as Hassler Whitney noted in 1932, duality is related to planarity. Whitney showed that “A graph is planar if it has a dual” (Wilson, etc.).

Constructing the dual of a graph helps to illustrate the important ideas relating to duality. First, label the edges and vertices of the graph noting the numbers of each component and each face. See Figure 2.20.

Figure 2.20. Stage I of Constructing the Dual of a Graph.
Now the dual can be constructed. Note: any new parts of a graph that are
drawn should be labeled to denote the dual—a complete and separate graph from the
original. The next step in constructing the dual is to draw a vertex inside each face, as
shown in Figure 2.21 (always including the infinite face) and label accordingly.

Third, connect the new vertices so that each existing edge is crossed exactly
once. If the graph is connected, as is Figure 2.20, then the connection of these vertices
in Stage III, likewise creates a unique graph known as the dual. As presented in
Wilson, there is a lemma which guarantees the uniqueness of connected graphs and
their duals. Figure 2.22 shows the unique dual.
Figure 2.22. Stage III of Constructing the Dual of a Graph.

The dual is complete. To make the distinction clear between the dual and the original graph, Figure 2.23 shows only the dual, the original graph has been eliminated.
It can be shown and proved with a lemma that the number of faces on the dual is the same as the number of vertices in the original graph. Likewise, the number of vertices in the dual is the same as the number of faces present in the original graph. In both the original graph and the dual the number of edges remains constant. In this case, both the graphs in Figure 2.20 and in Figure 2.23 the number of edges is six.

2.8.2 Isomorphisms

A type of a relationship of one graph with another is an isomorphism. An isomorphism describes any two objects of the same (iso) form (morph). In mathematics, two graphs have the same form if certain criteria are met. The criteria for two graphs to be isomorphic is the existence of a one–to–one correspondence
between vertices, which preserves edges. In other words, the adjacency of all vertices is preserved because corresponding edges connect corresponding adjacent vertices. The correspondence of vertices is determined by a function, \( f \). Figure 2.24 shows two isomorphic graphs and the functions relating each vertex.

![Figure 2.24. Isomorphic Graphs](image)

2.9 Conclusions

Chapter 2 introduces the basics of graph theory by presenting and defining terminology. These concepts are used in Chapter 3 to present some applications of graph theory and in Chapter 4 as the concepts used in the spiral curriculum.
CHAPTER 3

APPLICATIONS OF GRAPH THEORY

3.1 Introduction

It is important in mathematics to understand theory. The previous section illustrates the theoretical basis of graph theory. In reality, however, theory is not much use unless it can be applied. This section will describe various applications of graph theory, drawing on the definitions of the previous section. Graph theory, like all fields of mathematics, is applicable to many different disciplines. (Would any math teacher say otherwise?) While there are numerous fields and applications to choose from, Chapter 3 will illustrate graph theory by discussing five disciplines: chemistry, biology, humanities, engineering, and discrete math.

3.2 Applications in Chemistry

Chemical compounds are denoted in two ways: a formula and pictorially. The picture, in order to be applicable, must show the structure of the compound. In illustrating the structure, the placement of elements and the type of bonds connecting them are displayed. At a glance, people knowledgeable in chemistry can identify a compound. Simple graphs enable people to communicate the structure and identity of
chemical compounds. Elements are the vertices of the graph and the edges represent the type of bond connecting the elements. Figure 3.1, shows the graph of the chemical compound for water.

This graph is typical of the graphs used in Chemistry. There are two vertices of degree one, each connected by a bridge to the hydrogen atom.

Graphs are important to chemistry for two reasons. First, it often occurs that two compounds may appear to be the same because they share the same formulas. Compounds of this type are called alkanes. Upon further inspection, however, the graphs are not structured the same. In other words, two compounds may have the same formula but their graphs are not isomorphic. Butane and isobutane are alkanes.
The formula shared by these compounds is \( \text{C}_4\text{H}_{10} \). From their respective graphs, however, it is observed that they are indeed different compounds.

![Butane and Isobutane](image)

Figure 3.2. Chemical Alkanes.

The second reason graphs are important to chemistry is in the identification of new compounds. When working with new compounds, a structure may arise that is isomorphic to previously known structures. Making connections with known compounds makes the identification of new compounds more efficient.

3.3 Applications in Biology

Graphs are important in biology because they demonstrate the relationships among species. In the so-called niche overlap graph, species are represented by nodes.
Each node is connected if the adjacent species compete for the same food source. In the niche overlap graph in Figure 3.3, the shrew and mouse compete for the same food source.

Niche overlap graphs provide a good summary of food-competition interactions. The niche overlap graph provides biologists with an effective pictorial representation of an entire ecosystem. From the graph, complex analysis and comments can be made on the quality of connections between nodes and why certain species are not connected.
3.4 Applications in the Humanities

3.4.1 Influence Graphs

Similar to niche overlap graphs, influence graphs show the relationships between people. In an influence graph, nodes represent people. Two nodes (or people) are connected if one person can influence another. Influence means that one person can cause another or others to think, feel, or act differently. As illustrated in the influence graph shown in Figure 3.4, directed edges indicate whether a person is influential on others, or is swayed by others.

Figure 3.4. An Influence Graph.
3.4.2 Sociograms

Similar to an influence graph is a sociogram. A sociogram is also a digraph that represents people with nodes. Edges, in contrast with an influence graph, denote only people’s social perceptions of others. For example, in a classroom, a teacher would submit a survey of three questions to the class. Questions include: (1) Who would you most like to go to a movie with? (2) Who would you like to lead a class project? (3) Who does the best work in school? Tabulating the results of these three questions then graphing them illustrates the range of students present in the classroom, from the highly sociable students to those students who have very few friends. Figure 3.5 displays an example of a sociogram. In this case, only one question was asked of the students: Who would you like to lead a class project?

Figure 3.5. A Sociogram
As seen in Figure 3.5, Sir Isaac Newton is isolated from everyone else in the class—no one would choose him to lead a class project. Knowing this, the teacher can take steps to help Isaac feel more part of the group [5].

3.5 Applications in Engineering

As stated at the beginning of this section, there are countless examples to choose from. Engineering is important because everything that is human-made must first be engineered. To best illustrate the use of graph theory in engineering applications I will discuss electrical circuits and spring assemblies.

3.5.1 An Application in Electrical Engineering

The representation of electrical circuits, both graphically and mathematically, is accomplished through graph theory. The graph may not resemble the actual circuit, yet the graph maintain the same connectivity. The graph is better suited for analysis because it is a model. It is always easier to analyze and make changes on a model. In the context of this thesis, when modeling electrical circuits, nodes represent connection points and edges represent resistors. The following explanation is based on the circuit in Figure 3.6.
With the complete circuit, we can focus on its quantities of interest: voltage and current. A voltage is associated with each of the nodes, and a current with each branch. These are dynamic quantities that require more than a simple graph representation. Matrices allow the entire circuit to be analyzed.

The first matrix is a simple node-incidence matrix used to record the connectivity of nodes and edges. Matrix 3.1 is the node incidence matrix, as defined in Chapter 2, corresponding to the circuit in Figure 3.6.
Getting deeper into the quantities of the circuit (graph) we can now describe the voltage drop across each edge. The voltage drop on each edge is equal to the difference in voltage between incident nodes. In other words, \( \Delta v = v_n - v_{n+1} \). In this context, the equations describing the voltage drops for the circuit in Figure 3.6 are as follows:

\[
\begin{align*}
\Delta v_A &= v_1 - v_2 \\
\Delta v_B &= v_2 - v_3 \\
\Delta v_C &= v_2 - v_3
\end{align*}
\]

This system of equations lends itself to being expressed using linear algebra. When these equations are written in vector–matrix form, the node incidence matrix is evident.

\[
\begin{bmatrix}
\Delta v_A \\
\Delta v_B \\
\Delta v_C
\end{bmatrix} =
\begin{bmatrix}
1 & -1 & 0 \\
0 & 1 & -1 \\
0 & 1 & -1
\end{bmatrix}
\begin{bmatrix}
v_1 \\
v_2 \\
v_3
\end{bmatrix}
\]

\[\Delta v = -Av\]
Matrix $-A$ is essentially a discrete gradient operator—the gradient of voltage. This matrix provides an exciting connection to calculus for two reasons. First, finding a discrete gradient operator from an incidence matrix is not necessarily sought out in this arena. Secondly, the connection between this discrete gradient operator and the types of changes studied in calculus is very strong. Studying changes in the discrete sense is a natural first step toward studying change over continuous domains' as is done in calculus.

Returning to the circuit, the drop in voltage results in a current on each edge. Ohm’s law states that the current of each edge is proportional to the edge’s voltage drop. The constant of proportion is $G_x = \frac{1}{R_x}$, where $R_x$ is the resistance of the edge. Ohm’s law relates the voltage drop across each resistor and the current through each branch. Thus in satisfying Ohm’s Law for the circuit in Figure 3.6, the following three equations are necessary:

$$
i_A = G_A \Delta v_A$$

$$
i_B = G_B \Delta v_B$$

$$
i_C = G_C \Delta v_C$$

This system of equations is described succinctly with the following vector–matrix equation:

$$
\begin{bmatrix}
i_A \\
i_B \\
i_C
\end{bmatrix} =
\begin{bmatrix}
G_A & 0 & 0 \\
0 & G_B & 0 \\
0 & 0 & G_C
\end{bmatrix}
\begin{bmatrix}
\Delta v_A \\
\Delta v_B \\
\Delta v_C
\end{bmatrix}

= \mathbf{G}(\Delta \mathbf{v})
$$

Matrix 3.3. Matrix Satisfying Ohm’s Law and the Symbolic Form for this Matrix.
The next property to consider is a conservation law known as Kirchoff’s Current Law. This law states that if any portion of the circuit (which means that the circuit could also be considered in its entirety) is analyzed, the current flowing in that point must equal the current flowing out. When analyzing electrical circuits, conservation of current is checked at each node. There are three nodes in the electrical circuit in Figure 3.6, so there are three equations satisfying Kirchoff’s Current Law.

\[
\begin{align*}
I_1 + i_A &= 0 \\
I_2 - i_A + i_B + i_C &= 0 \\
I_3 - i_B - i_C &= 0
\end{align*}
\]

These equations, like all others presented in this section, are easily written in matrix form:

\[
\begin{bmatrix}
I_1 \\
I_2 \\
I_3
\end{bmatrix} =
\begin{bmatrix}
-1 & 1 & 0 \\
0 & -1 & 1 \\
0 & -1 & 1
\end{bmatrix}
\begin{bmatrix}
i_A \\
i_B \\
i_C
\end{bmatrix}
\]

\[
\mathbf{I} = \mathbf{A}^T \mathbf{i}
\]

Matrix 3.4. Matrix Satisfying Kirchoff’s Current Law and the Symbolic Form for this Matrix.

The coefficient matrix of Kirchoff’s Current Law is the transpose of the node incidence matrix. So, as matrix $-\mathbf{A}$ is a discrete gradient operator, matrix $\mathbf{A}^T$ is a discrete divergence operator. With this connection, it is easy to combine these systems of equations by substitution.
\[ \Delta v = -A v \]
\[ i = G(\Delta v) = -G A v \]
\[ I = A^T i = -A^T G A v \]

The last of the three equations relates the input currents, I, to the node voltages, v, in the circuit. These combined equations provide an example of the work done by electrical engineers. With these three equations, engineers are equipped to analyze more complex electrical systems.

3.5.2 An Application in Engineering Mechanics

Structures and mechanics can be modeled by assemblies of springs. Systems of elastic springs parallel the relationship described in circuits in the sense that the mathematics is the same, but the actual quantities studied are different. Instead of using matrices to study current and voltage, civil and mechanical engineers analyze forces and displacement of structures and mechanisms. Setting up the matrices and performing the mathematics follows the same sequence as in an electrical circuit.

Figure 3.7 displays the assembly of springs used for this discussion.
The first characteristic to describe is the connectivity of the springs. This information is recorded in a node incidence matrix, $A$.

$$
A = \begin{bmatrix}
-1 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 \\
0 & 0 & -1 & 1
\end{bmatrix}
$$

Matrix 3.5. Node Incidence Matrix for Figure 3.7.

The node incidence matrix provides a key ingredient for describing the next characteristic of the springs: the kinematics of the system. The kinematic relationships
in the assembly describes the stretch in each spring in terms of the joint displacements.

So, the stretch in a given spring, \( \Delta_n \), can be calculated as \( \Delta_n = u_{x+1} - u_x \). The equations describing the stretch on each spring are:

\[\begin{align*}
\Delta_A &= u_1 - u_2 \\
\Delta_B &= u_2 - u_1 \\
\Delta_C &= u_3 - u_2
\end{align*}\]

These equations can be expressed via matrices.

\[
\begin{bmatrix}
\Delta_A \\
\Delta_B \\
\Delta_C
\end{bmatrix} =
\begin{bmatrix}
-1 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 \\
0 & 0 & -1 & 1
\end{bmatrix}
\begin{bmatrix}
u_0 \\
u_1 \\
u_2 \\
u_3
\end{bmatrix}
\]

\[\Delta u = Au\]

Matrix 3.6. Matrix of Joint Displacements and the Symbolic Form for this Matrix.

Matrix \( A \) is not merely a node incidence matrix. This matrix is connected to calculus as a gradient operator, just as is the case for the node incidence matrix describing the connectivity in the electrical circuit.

The third characteristic to describe relates the deformation of the springs to the forces in them. Each spring obeys Hooke’s Law which states the force in the spring is proportional to the stretch. The constant of proportionality is called the stiffness of the spring and it specifies the amount of force required to stretch a spring one unit. The relationship is described, mathematically, as \( F = k\Delta \). In this application, three equations are necessary to describe Hooke’s Law for each spring.
$F_A = k_A \Delta_A$

$F_B = k_B \Delta_B$

$F_C = k_C \Delta_C$

Hooke's Law is analogous to Ohm's law used in electrical circuits. Following the example of Ohm's law for the electrical circuit, Hooke's Law can also be described with a matrix-vector equation.

\[
\begin{bmatrix}
F_A \\
F_B \\
F_C
\end{bmatrix}
= \begin{bmatrix}
k_A & 0 & 0 \\
0 & k_B & 0 \\
0 & 0 & k_C
\end{bmatrix}
\begin{bmatrix}
\Delta_1 \\
\Delta_2 \\
\Delta_3
\end{bmatrix}
\]

\[\mathbf{F} = \mathbf{k} \Delta\]

Matrix 3.7. Matrix Satisfying Hooke's Law and the Symbolic form for this Matrix.

The next step is to describe the forces that are acting on each joint. These forces must balance; in other words, there is an equilibrium of forces. Figure 3.8 shows each node individually with the equations describing the equilibrium of the forces. $F_N$ represents the force in each spring, $N$. $F_j$ is the force in the joint from an external agent.
Figure 3.8. Forces Acting on each Joint and the corresponding Equilibrium Equations.

These equations are easily put into matrix form.

\[
\begin{bmatrix}
P_0 \\
P_1 \\
P_2 \\
P_3 \\
\end{bmatrix} =
\begin{bmatrix}
-1 & 0 & 0 \\
1 & -1 & 0 \\
0 & 1 & -1 \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
F_A \\
F_B \\
F_C \\
\end{bmatrix}
\]

\[
P = A^T F
\]

To describe these three equations, the same incidence matrix, $A$, is present in the first and third matrix; it is simply transposed. As was the case in the electrical circuit, matrix $A$ is the gradient operator and $A^T$ is the divergence operator, providing a connection to calculus.

These matrices enable easy substitution, allowing engineers to write a relationship between $P$ and $u$.

$$\Delta = Au$$
$$F = k\Delta = kAu$$
$$P = A^T F = A^T kAu$$

In the last equation, $u$ can be determined if $P$ is given.

Electrical circuits and assemblies of springs provide excellent examples of engineering applications of graph theory. These two applications are doubly interesting because they provide a bridge between discrete mathematics and calculus via the discrete gradient and divergence operators.

3.6 Applications in Discrete Mathematics

There are many applications of graph theory in the area of discrete mathematics. Graphs, by definition, are discrete structures; it follows, logically, that discrete structures are used in discrete mathematics. The discrete structures are no longer termed “graphs.” Technically, discrete applications use graphs, but when we are operating in the “real world”—pure applications—rather than under a theoretical base, the terms change. Now, instead of applying graphs, we will use networks. A network is a weighted graph. In section 3.5, networks were used in the engineering
applications. Weighted edges can be length, cost, capacity, resistance [7], or spring stiffness—depending on the particular application.

The possible applications to illustrate are countless. This paper, however, will describe three applications: (1) assignment problems, (2) minimum cost spanning trees, (3) cost-flow networks. These three applications are used because they are giants of graph theory and, together, they illustrate their power and versatility.

3.6.1 Assignment Problems

People must make assignments everyday. A typical example involves 15 of your closest friends, and you, attempting to go to a movie with the limitation of having only four cars. You must decide who travels in which car. In other words, your task is to assign people from group I (your 15 close friends) to group II (the four cars). Any task involving the matching of members of two distinct groups is an assignment problem. Other examples include matching students with classrooms, loading seven different satellites into two cargo bays of a space shuttle, and pairing people off for marriage. Bipartite graphs model assignment problems.

As defined in the previous section, bipartite graphs partition the vertices into two nonempty groups, such that any edge connects a vertex from each group. Applying this definition to the movie example above, the friends represent one group and the cars the other. The edges connecting these two sets represent who will ride in which car.

To explain how to solve assignment problems, most authors use the example of matching couples. Marriage is the perfect example of assignment applications because this institution is so familiar to us. Besides, this is yet another example of how mathematics applies to our every-day lives. As Strang explains, “you may think that marriage is outside the scope of applied mathematics. That is true. Once you are
married, mathematics is of no possible use. But there is a superficial model for a simpler problem—getting married in the first place” [7]. Marriage is the context for explaining how to solve assignment problems. It is important to remember, however, that the context of the problem is not the key point. The emphasis is on the process that is utilized to solve assignment problems.

This application is modeled after a similar application found in Introduction to Applied Mathematics by Strang. To begin, we have two, distinct, nonempty sets: (1) Hypatia, Maria Agnessi, Emmy Noether, and Sophie Germain; (2) Leonhard Euler, Carl Friedrich Gauss, Sir Isaac Newton, and Leonardo Fibonacci. (This is, of course, no reflection on history.) Our brides-to-be are a discriminating set. They won’t marry just any of the four choices. Matrix 3.9 represents the pairing preferences. The rows of \( A \) are the women, while the columns are the men. A “1” in position \( ij \) represents a willingness of woman \( i \) to marry man \( j \), while a “0” means otherwise.

\[
A = \begin{bmatrix}
Hypatia & Euler & Gauss & Newton & Fibonacci \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0
\end{bmatrix}
\]

Matrix 3.9. Matrix of Preferences.

Figure 3.9 is the bipartite graph corresponding to the preferences expressed in Matrix 3.9.
The first step in assigning marriages is to determine the maximum number of assignments that are possible. The answer to this question is contained in matrix $A$.

"The maximum number of marriages equals the minimum number of lines (rows or columns or a combination of both) that contain all the 1's in the matrix" [7]. The minimum number of lines needed to contain all the 1's of Matrix 3.9 is three, as shown in Matrix 3.10.
Of the possible three marriages, Hypatia could marry any of the eligible bachelors. Agnessi’s only option is to marry Fibonacci, while Germain’s only option is Gauss. Noether could marry either Gauss or Fibonacci.

The next step is to determine which two of the possible eight will not be happily joined in marriage. Again, the information is contained in the matrix. The vertices covered by a line determines which six will be joined. Any “1” covered by a line is a possible marriage. The person making the assignments may choose the “1’s” in an optimal way as long as this condition is met: each vertex of the graph can only appear once. By convention, once a match is chosen, it is circled. Once the three marriages are chosen, it is easy to check that the condition is met. Each row and column of the matrix can only contain one circle. The circled vertices are called the cover of the matrix. Matrix 3.11 shows a possible way of making the marriage assignments.
It should be noted that Hypatia (row 1) could have actually married either column 1 or column 3. It doesn’t matter which assignment was made because she maintains an equal preference for either 1 or 3 and choosing either one will still maintain the cover. (Now we know the real reason Newton remained a Bachelor.) Relating the cover of Matrix 3.11 with the original bipartite graph shows which couples will be united in marriage.
The three marriages, as shown by the edges, represent the matching set. Edges that have no vertices in common constitute a matching set because they meet the condition of only appearing once. In the context of this problem, this means bigamy is not allowed. The matching set contains three edges, although a maximum of four is possible. Therefore, this is not a complete matching. To be complete, the number of edges in the matching set must equal the maximum allowable assignments. The maximum assignments is determined by the minimum number of rows or columns.

The vertices, together, comprise the set that meets every edge of the matching. This set of vertices is called the cover. Figure 3.10 shows the cover is comprised Hypatia, Agnessi, Noether, Euler, Gauss, and Fibonacci. The matching and the cover are related by the König–Egerváry theorem which states:
The maximum number of edges in a matching equals the minimum number of vertices in a cover.

In the context of this assignment problem, the $\text{max}(\text{matching}) = \text{min}(\text{cover}) = 3$.

To review, there are four steps to make assignments:

1. Construct the bipartite graph and the corresponding matrix.
2. Cover all the 1's of the matrix with the minimum number of lines. The minimum number of lines equals the maximum number of possible assignments.
3. Find the matching by choosing the 1's covered by a line, and satisfying the condition that only one choice is allowed per row and per column.
4. Find the cover by finding the set of vertices that meet every edge.

### 3.6.1 Minimum Cost Spanning Trees

In most applications, every node needs to be connected to the network, it is not necessary, however, for each node to connect with every other node. Connecting each node with every other node is impractical and redundant. There is a cost incurred with each connection—for example, cost per length of cable—which dictates which connections will be made.

Applications requiring minimum spanning trees include connecting cities with a computer network, connecting computers on a college campus, and making connecting flights while traveling across country. In many cases, the type of algorithm that works best is a greedy algorithm. These types of algorithms dictate that at each step of constructing the spanning tree, one should make the optimal choice by choosing a least weighted edge, or arc. Greedy algorithms will often create the minimum spanning tree—a tree connecting every node with the smallest sum of edges.
The two best known greedy algorithms that produce minimum spanning trees are Prim's and Kruskal's Algorithms. To illustrate the use of these two algorithms, seven computers, located in different cities, must be connected into one network. The weight of the edges might represent the distance between cities to the nearest 10 miles. Figure 3.11 shows the network of computers.

![Figure 3.11. A Computer Network.](image)

Prim's algorithm will yield a single tree. To use Prim's algorithm:

1. Determine the number of nodes, \( n(v) \).
2. Arbitrarily choose an edge of minimum weight.
3. Add the shortest edge incident with a vertex already in the tree and not forming a cycle.

4. Continue adding the edge of least weight incident with the existing tree until $n - 1$ edges are added.

In the example from Figure 3.12, $n(v) = 7$, so the resulting minimum cost spanning tree will have six edges. There are three edges equal to the minimum weight of one: $\{A, B\}$, $\{B, D\}$, $\{C, D\}$, and $\{D, F\}$. Arbitrarily choose $\{A, B\}$ as the first edge. Five more edges must be added to create the minimum cost spanning tree. In this case, as in most applications, there is not one unique answer. What follows is the construction of one possible spanning tree:

![Diagram of minimum cost spanning tree construction](image-url)
\[
\{D, F\} = 2
\]

\[
\{C, F\} = 1
\]

\[
\{D, E\} = 2
\]
Figure 3.12. Minimum Cost Spanning Tree via Prim’s Algorithm.

The spanning tree in Figure 3.12 has a sum of nine.

Kruskal’s algorithm, on the other hand, will find the minimum cost associated with a particular application by finding the minimal spanning forest (a set of trees). In other words, several trees are allowed to grow at once, connecting into a single tree in the final step. Using Kruskal’s algorithm involves the following steps:

1. Determine the number of nodes, \( n(v) \).
2. Order the edges according to weight.
3. Add edges of least weight, according to the ascending order determined in Step 2, until \( n - 1 \) edges are added. Do not add an edge if it will create a loop.
Kruskal’s algorithm is now used to find the shortest path for the computer network in Figure 3.12. The number of nodes, again, is $n(v) = 7$. The edges are ordered alphabetically by weight.

<table>
<thead>
<tr>
<th>=1</th>
<th>=2</th>
<th>=3</th>
<th>=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>${A, B}$</td>
<td>${D, E}$</td>
<td>${A, C}$</td>
<td>${F, G}$</td>
</tr>
<tr>
<td>${B, D}$</td>
<td>${D, F}$</td>
<td>${G, D}$</td>
<td></td>
</tr>
<tr>
<td>${C, F}$</td>
<td>${E, G}$</td>
<td>${D, E}$</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1. Ordered Edges by Weight—Step 2 of Kruskal’s Algorithm.

Begin by adding all the edges with weights equal to one that do not create cycles. Now add edges of weight equal to two, three, and four, again avoiding any cycles, until six edges are added. Using the edges, as ordered in Table 3.1, the following tree can be constructed.

```
A
  |__
  |
B
```
\{B, D\} = 1

\{C, F\} = 1

\{D, E\} = 2

\{D, F\} = 2
None of the edges with weights of three or four are used to create the minimum spanning tree. This tree yields a sum of nine. In this case, the tree created using Kruskal’s Algorithm is the same as the tree found in Prim’s algorithm. Similarly to Prim’s algorithm, the tree found with Kruskal’s algorithm is not unique. If the edges had been ordered differently in Step 2—for example, not alphabetically—then the edges added during Step 3 might have been different.

The example used to illustrate Prim’s and Kruskal’s algorithm is very simple. Only seven cities were part of the network. Quite a bit of space and mental processing time is required to find the minimum cost spanning trees. Computers enable Prim’s and Kruskal’s algorithm to be iterated quickly and efficiently, provided the number of...
vertices is not too large. These two algorithms are relatively easy to program; thus, using a computer, these algorithms are ready tools for many applications requiring a minimum cost spanning tree.

3.6.3 Cost Flow Networks

Networks are very useful in making decisions; especially decisions involving many choices. Once these choices are modeled with a graph, an algorithm can often be found that makes the best, or close to the best choices.

Most contexts call for specific types of choices. For example, a company will want to maximize profits; maximize the amount of oil flowing through a pipeline; or minimize distance traveled during a sales trip. One algorithm that solves problems of this type is the Ford–Fulkerson method.

To demonstrate the Ford–Fulkerson algorithm, the revenues of Saint–Backs, a company that produces book covers for college students will be maximized. This example is modeled after a cost–flow network found in Winston’s *Operations Research: Applications and Algorithms*. Saint–Backs have several different options for shipping their product from their warehouse to a college. Each shipping option is limited to a maximum carrying load. The goal is to maximize the amount of book covers that can be shipped, thus maximizing revenue. In maximizing revenue, this example assumes that every book cover shipped will be sold. Figure 3.14 depicts the network described in the Saint—Backs application.
The warehouse of book covers is labeled $s_0$, for the source; and the college to receive the shipment is $s_0$ or the sink. The arc labeled $a_o$ is simply a bookkeeping tool, it is not actually part of the Ford–Fulkerson model. This arc records the maximum flow of the network as determined by the maximum capacity of the edges flowing directly into the sink. In this example, the maximum number of shipments the Saint–Backs can send, via any option, is three because the two arcs flowing directly into the sink have a combined weight of three.

The first step in finding the maximum flow of a network is to find a feasible flow of the network. Simply find a path that will allow flow through each edge, resulting in flow reaching the sink. Logistically, to record this initial feasible flow, write next to each weight the flow in parenthesis. For example, an arc using three out of a capacity of five is labeled $5(3)$. Five units can be sent, but only three are used. This initial flow
is the first draft, so to speak, which will enable a maximal flow to be found. Figure 3.15 shows the “first draft” for the Saint—Backs example. The flow entering the sink is two, so $a_0=2$.

Next categorize the flow of each node in one of three ways:

I— if the flow of the arc can be increased (the weight in the parenthesis is less than the maximum capacity of the arc);

R— if the flow of the arc can be reduced (the weight in the parenthesis is greater than zero); and

I/R— if the flow of the arc can be either increased or reduced (the
weight in parentheses is not equal to zero or the maximum weight possible).

Table 3.2 summarizes the first two steps of the Ford–Fulkerson Algorithm as it applies to the Saint–Backs example.

<table>
<thead>
<tr>
<th>ARC</th>
<th>FEASIBLE FLOW</th>
<th>TYPE OF FLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀—1</td>
<td>2(2)</td>
<td>R</td>
</tr>
<tr>
<td>s₀—2</td>
<td>3(0)</td>
<td>I</td>
</tr>
<tr>
<td>1—2</td>
<td>3(2)</td>
<td>I/R</td>
</tr>
<tr>
<td>1—3</td>
<td>4(0)</td>
<td>I</td>
</tr>
<tr>
<td>2—s₁</td>
<td>2(2)</td>
<td>R</td>
</tr>
<tr>
<td>3—s₁</td>
<td>1(0)</td>
<td>I</td>
</tr>
</tbody>
</table>

Table 3.2. Steps 1 and 2 of the Ford–Fulkerson Algorithm.

Once the flow is categorized, the next step is to find an optimal path through the network from the source to the sink. This path will determine whether to increase or decrease flow through which nodes and by how much. Finding the path is a three step process.

1. Label the source $[1, \infty]$.

The first position is blank because it is unimportant where the flowing material came from—it is in the network and that is all the needed information. Since it is not known where the flow came from, the bounds (if any) are not known; so the source is assumed to be boundless.
2. Label the edges and nodes.

Assign to each edge the maximum allowable flow. This assignment is made according to conservation laws: flow in to each node equals flow out. It is possible for flow to be less than maximum, but never more. The nodes regulate the flow. Begin with the source, then move to the next adjacent node via an arc whose capacity is not met. In other words, an arc that is a member of I or I/R. Label each node encountered. The label consists of two pieces of information: the previous node and how much more that node can send to meet the capacity of the arc. For example, if the first node labeled from the source is node 1 describing an arc with capacity of 5(3), the label is [so, 2] because the source can send two more units on this arc to meet the capacity of five. Label each arc of the path in one of two ways:

F – if the arc is a forward arc. The nodes follow in a sequential order and the arc is a member of I or I/R; or

B – if the arc is a backward arc. The nodes follow in descending order and the arc is a member of R or I/R.

Figure 3.16 shows the labeled Saint—Backs network.
Table 3.3 summarizes the labeled Saints—Back network.

Table 3.3. Labeled Edges and Nodes for the Initial Flow.
The subscript attached to the "F's" and "B's" denotes the order in which the edges were added in Step 2 of analyzing the "first draft" of the network. Only four edges are used to reach the sink—these four edges constitute a path.

3. Determine if the flow is maximal.

If the sink is reached via the labeling process, the flow is not maximal.

Flow must be increased. Using the information regarding the nodes used in the path reaching the sink, determine how much to increase the flow. This is a two step process. First find the minimum of all the edges used in the path that are labeled I.

\[ \min(3, 4, 1) = 1 \]

Do the same for the arcs used in the path that are labeled B. (If the path created contains only forward arcs, omit this step.)

\[ \min(2) = 2 \]

Now find the minimum of the two minimums.

\[ \min(1, 2) = 1 \]

This amount tells how to change the capacity of the flow. Increase the capacity of all forward arcs by this amount while reducing the backward arcs by the same amount.

Figure 3.17 shows the adjusted flow through the network.
4. Stop when $a_o$ is reached.

Or, when the sink is unable to be labeled.

As seen in Figure 3.18, $a_o=3$, so the flow is maximal.

To check if the flow is at maximum, check the cuts of the network. A cut is the removal of any edge or edges that disconnect the sink from the source. The capacity of the cut measures the maximum allowable flow of the network. According to the Max flow–Min cut theorem, as presented by Winston, finding the minimum capacity cut, finds the maximum flow of a network. In the Saint—Backs network, the minimum cut is comprised of edges 2—$s_f$ and 3—$s_i$, which yields a capacity of three.
Chapter 3 explains five applications of graph theory. Again, there are numerous applications in varied disciplines. These particular applications were chosen to illustrate the strengths of graph theory. These strengths are used in the application of education in the following chapter.
CHAPTER 4

THE SPIRAL CURRICULUM

4.1 Introduction

This chapter describes one of the most important applications of mathematics: education. Educating the next generation to be mathematically literate is critical. Consequently, much thought and analysis is required before the actual teaching can begin. This "preparatory" time allows teachers to plan two things. First of all, forethought enables teachers to prepare a curriculum. As defined by the National Council of Teachers of Mathematics [NCTM]

A curriculum is an operational plan for instruction that details what mathematics students need to know, how students are to achieve the identified curricular goals, what teachers are to do to help students develop their mathematical knowledge, and the context in which learning and teaching occur [10].

The curriculum provides the instructional framework for teachers. Education occurs within this global structure.

The second benefit of teacher planning is more specific than the global curriculum. Implementing the curriculum on a daily basis requires daily plans. Thus, these specific daily plans are the second result of preparatory time.
Operational plans of individuals—teachers, schools or school districts—can vary widely. In the context of this thesis, the curriculum will apply to one school system; specifically, a system that includes students ranging from ages six to eighteen (analogously, grades kindergarten through twelve). The curriculum plan that will be developed is created by the author for the purposes of this thesis and to illustrate a particular type of curriculum. In reality, a curriculum that spans the ages of students discussed in this paper would be created by a team and implemented on a school-wide or district-wide basis.

Educational research provides several frameworks for curriculum planning. Each curriculum has pros and cons. Sorting through these positives and negatives enables educators to choose a curriculum which can lead to quality education.

Curriculum plans reflect different philosophies of education; it follows, then, just as there are different philosophies of education, there are different curriculums which reflect each philosophy. For example, Mr. Al G’Rithim is a fifth grade math teacher who espouses beliefs that students need only know “the basics.” This translates to mean that students learn algorithms and spend time in drill and practice, with little emphasis on application. Mr. G’Rithim follows the district approved textbook, which includes a prescribed course flow, to the letter, allowing no flexibility for variability among student learning rates, learning styles, or learner interests. Math is taught in an authoritative manner; students are passive and learn everything from Mr. G’Rithim who is the only authority of mathematics in his classroom. Mr. G’Rithim’s philosophy of education can be summarized as authoritative, inflexible, unconnected with information (mathematical, or otherwise) that his students have gained in previous years, will learn in future classrooms, or with applications to life-experiences that take place outside his classroom. Consequently, Mr. G’Rithim’s curriculum plan which will...
put his philosophy into action will consist of book work emphasizing drill and practice (memorization) rather than a true understanding of mathematics.

Mr. G'Rithim is introduced in order to illustrate the point that an individual's philosophy of education is important to know because each philosophy determines what sort of decisions will be made regarding a curriculum. It is not the purpose of this thesis to discuss different philosophies of education. The purpose is to explain and demonstrate a particular curriculum, which necessitates an explanation of a particular philosophy of education. It should be noted that planning the mathematics curriculum to coincide with a philosophy of education is labor intensive. The costs of the labor are easily offset, however, by the benefits of providing quality education for students. Ethically, providing the highest quality of education to students is the prime directive to all teachers.

The philosophy of education in this thesis is best described as Developmental. Philosophies that are developmental in nature are closely linked to developmental psychologists such as Jean Piaget. In brief, Piaget believes that human beings maintain two basic tendencies: organization—"tendency to systematize processes"—and adaption—"tendency to adjust to environment" [1]. Organization and adaption are "invariant functions" [1]. Neither organization or adaption vary with age. In other words, an infant uses these thought processes the same way an adolescent does, which is the same way an adult thinks. Schemes, on the other hand, are not invariant. The "organized pattern of behavior or thought" of a human as "they interact with their environment, parents, teachers, and agemates" are [1]. The schemes "undergo systematic change at particular points in time" [1]. Common sense tells us that a three-year-old obviously thinks differently than a thirty-year-old. The changes that Piaget refers to occur at four stages of development. The particular stages are not critical to note; it is important, however, to understand that
children do not suddenly 'jump' from one stage to the next. Their cognitive development follows a definite sequence, but they may occasionally use a more advanced kind of thinking or revert to a more primitive form. The rate at which a particular child proceeds through these stages varies, but Piaget believes the sequence is the same in all children [1].

An interesting question resulting from Piaget’s work is related to grouping children for educational purposes. If the rate children master each stage varies, why are children grouped according to age? Age would seem to be irrelevant to a child’s readiness to learn. Why not group children according to what “stage” they are in? After all, there is no magical equation relating the number of days a person has been on this planet to suddenly having the ability to perform calculus. Answering this question could be the topic of another thesis.

Historically, after Piaget came an educational psychologist named Jerome S. Bruner. Bruner, like Piaget, believed in a developmental process of educating children. Bruner was a proponent of two educational ideals that are very important to math education. First, Bruner was a proponent of discovery learning. When a student experiences discovery learning, the student is an active participant in the learning process. Instead of the teacher taking on the role of the all-knowing dispenser of knowledge with passive students that simply absorb whatever the teacher is spouting, discovery learning requires that the students take responsibility and control of their own learning. The role of the teacher is to guide students through a learning situation, usually via questioning. Students explore, question, investigate, and delve into whatever topic is put in front of them. The rationale behind discovery learning is that the teacher leads or guides an individual to discover new information. Once the students “discover” an idea it is theirs to own. No one simply told them to memorize an
idea or relationship, through their own skills (and the skills of the teacher) the student develops an understanding of an idea. The NCTM also supports discovery learning as a key component of mathematics curriculums.

Another major contribution made by Bruner is his unshaking support in the concept of a spiral curriculum. Bruner, as chair of the Woods Hole Conference which took place in 1959 (a national conference sponsored by the federal government; the purpose of the conference was to discuss education reform [3]) formally describes a spiral curriculum as

the form of a metaphoric spiral in which at some simple level a set of ideas or operations were introduced in a rather intuitive way and, once mastered in that spirit, were then revisited and reconstrued in a more formal or operational way, then being connected with other knowledge, the mastery at this stage then being carried one step higher to a new level of formal or operational rigor and to a broader level of abstraction and comprehensiveness. The end state of this process was eventual mastery of the connexity and structure of a large body of knowledge [3].

The spiral curriculum will provide the curricular framework for teaching graph theory. So, graph theory in its most fundamental sense is introduced to children at the elementary level, and these ideas are revisited each year until as an undergraduate, the students are applying graph theory in all its sophistication.

In a similar vein to Bruner’s Spiral Curriculum, the NCTM establishes five goals for all students to achieve mathematical literacy. These goals are the same for beginners—elementary grades—and students with advanced skills—undergraduates. The approach or emphasis on particular math content changes as the students gain mathematical sophistication; however, the goals are present at each level. In other words, the goals do not change, simply the expression or teaching of the goals changes.
In education, the goals are expressed and spiraled through the curriculum. According to the NCTM, the goals supporting the curriculum are:

1. Students learn to value mathematics.
2. Students are confident in their ability to do mathematics.
3. Students become mathematical problem solvers.
4. Students communicate mathematically.
5. Students learn to reason mathematically.

To make these goals easier to implement, the NCTM divides students into three categories: Beginner (kindergarten through fourth grade), Intermediate (fifth through eighth grade), and Advanced (ninth through twelfth grade). The categories do not necessarily coincide with grade levels. The NCTM goals are developmental. Grades have been included because this is how our current system of education operates. Please note that when grades are included, they are only an approximation of where students would fit, developmentally, into the NCTM categories.

4.2 Modules

What follows this introduction are five modules. Each module contains six pieces of information. (1) Lesson Identification. This simply identifies the sophistication level of the spiral and the name of the unit. (2) General Objectives. These objectives target what students will learn during the unit. (3) Specific Objectives. These are the specific activities necessary for the students to reach the General Objectives. (4) Rationale. This section provides a reason why this lesson is important to the students. (5) Content. The Content outlines the activities of the
teachers and the students—these are essentially daily lesson plans. (6) Relation to
NCTM Goals. This final section addresses how each module fits into the spiral.
I. Lesson Identification

GRADE: 2

TOPIC: Which Way Do We Go?

II. General Objectives

Nature & Scope
Students will use principles of graph theory to create, read, and follow maps. Students will gain personal skills and an understanding of the applications of math.

Major Outcomes
1. Students know the definition of a simple graph.
2. Students display drawing skills.
3. Students relate actual distance to scale distance on paper.
4. Students use ordinal directions.
5. Students recognize the significance of maps.

III. Specific Objectives

1.1 Students write a legend to their map correctly defining all vertices and edges.
1.2 Using graph paper, students draw vertices at intersection points.
2.1 Students connect vertices with a straight edge.
3.1 Using graph paper, students equate one square to an appropriate number of paces (2nd-grade-size steps).
4.1 Students construct maps in relation to ordinal directions.
4.2 Students follow a map correctly using ordinal directions.
5.1 Students write three sentences explaining the importance of maps.

IV. Rationale

Maps are recognized for their importance to human survival. The historical basis of maps is indisputable. This unit will help students understand how a map is created and how to use a map to get from location to location. Maps provide an excellent vehicle to connect mathematics to everyday realities and demands of our world.
V. Content

Day 1
Set
- Activate background knowledge via questions.

How did pirates find their treasure?

Body
- Explain “Rationale”; introduce major activity of unit: Making a Map.
- Brainstorm as class:
  1. Places to go?
  2. Where do you find maps?
  3. Focus question:
     Different places to visit around a town?
- Using overhead: Look at the simplified (all streets are straight and intersect at right angles) map of a town
  1. Identify parts of a map: legend, ordinal directions, introduce the idea of scaling.
  2. Introduce graph terminology:
     - Lay a clear transparency over the map.
     - Pick two locations to travel between (make sure they include at least two turns, or changes in direction, en route).
     - Trace path via the streets.
     - Remove map to reveal simple graph: vertices, edges, path.

Closure
- Students write names of the parts of the graph: vertex and edges.

Day 2
Set
- “Game Show.” Review yesterday’s terminology via question/answer; throw out prizes for correct answers. Terms: map, legend, ordinal directions, vertex, edge, and path.

Body
- Review the “Rationale” for the unit.
- Using overhead: display a map of the school—discuss scale.
  1. What are distances on the map?
  2. Why is the map smaller than actual distances?
- Field trip: Identify locations on map and pace off distances between points.
  1. List distances on board.
  2. Teacher models scaling distances the number of squares on paper.
- Students create own maps of school; including a legend identifying all vertices, distances (in steps) between vertices, and the ordinal directions.

Closure
- Students write three sentences regarding the importance of maps.
VI. Relation to NCTM Goals

1. **Valuing Mathematics**—The students recognize maps as tools used in everyday life. Knowing the mathematical basis of maps, specifically graph theory, the students are learning the value of mathematics as a tool.

2. **Confidence in Math Ability**—Confidence, in an area, arises primarily from achievement. As Jeff Howard, President of the Efficacy Institute in Lexington, Massachusetts explains:

   Development is fueled by effective effort, that is, a committed, focused, and strategic approach to work. The capacity to commit effort of this kind is a function of children's psychological states, especially their confidence that they have what it takes to learn [4].

   The more a student achieves, the more confidence is gained, which in turn fuels more effort, which logically leads to more achievement. It is a cycle that positively impacts the development of students. This unit does foster achievement for students. They develop a map of their school which includes specific terminology and ideas.

3. **Becoming Mathematical Problem Solvers**—The problem solved in this situation is scaling. Students must scale distances to fit a map, while maintaining a correct representation of the actual building. In solving this problem, students learn that graphs are excellent models of the real world.

4. **Communicating Mathematically**—Students communicate mathematically in two ways. First, they must create a mathematical model in the form of a map. Second, they write in their journal about the importance of their model.

5. **Reasoning Mathematically**—Students must organize and synthesize dimension into their mathematical model.
I. Lesson Identification

GRADE: 5

TOPIC: What's Your Type?

II. General Objectives

Nature & Scope
This unit, specifically, will enable students to recognize types of graphs and their applications. In general, “What’s Your type?” fosters students' math literacy by giving them practice in identifying and manipulating patterns.

Major Outcomes
1. Students know different types of graphs.
2. Students identify isomorphic graphs.
3. Students create duals of graphs.
4. Students recognize the significance of graphs.

III. Specific Objectives

1.1 Students list the characteristics of the following graphs: simple, bipartite, trees, stars, and multigraphs.
1.2 Students construct the graphs listed in 1.1.
2.1 Students match isomorphic graphs.
3.1 Students list the steps to create a dual.
3.2 Students construct the dual of graphs.
4.1 Students write two paragraphs explaining the application of graphs in modeling situations in every-day life.

IV. Rationale

It is important and helpful to describe a situation in several different ways—orally or in writing. Sometimes writing is not enough. A picture or mathematical model of a situation complements a description; often, the model is sufficient to stand alone as the explanation. Graphs provide useful and excellent models.
V. Content

Day 1
Set
• “What is this?” Provide map of a city—the map is a model.
Body
• Explain “Rationale.”
• Review components of a graph: vertex and edges.
• Making a model:
  1. Teacher explains how the city map is a model of the actual city—also, it is a simple graph.
  2. Students must create models using only graphs.
  3. In groups of four, the students model the following situations:
     = A computer network—wheel,
     = Water sources and lines connecting to houses—bipartite.
     = Various roads, or paths, used to navigate through a town—multigraph.
     = Airplane flight paths—digraph
     = A basketball tournament—tree.
  4. Groups present and defend their models.
• Teacher summarizes the five types of graphs presented; including the qualities and characteristics that differentiate the types.
Closure
• Students create a written summary (list or chart) organizing the qualities of the different types of graphs.

Day 2
Set
• “Beat the Clock!” On scratch paper, students list pairs of objects with the same form in a given amount of time. Prizes are given to the students with the most pairs.
Body
• Review the “Rationale” for the unit.
• Define: ISOMORPHISM and how this relates to modeling—sometimes different forms of the same shape provide a more efficient model.
• Via question/answer, class identifies characteristics of a graph: the number of edges, the number of vertices, the degree of each vertex.
• Teacher models how to find an isomorphism.
• Students practice finding isomorphic graphs in pairs.
Closure
• Students write an example of a situation, besides graphs, requiring an isomorphism.

Day 3
Set
• Using magnets, the teacher demonstrates repelling and attracting.
  The point: Sometimes an opposite perspective is necessary, just as sometimes we need a different perspective of graphs.
Body
• Review “Rationale.”
• Define: DUAL and how this relates to modeling.
• Identify characteristics of a graph necessary to create a dual: edges, vertices, and faces.
• "You make the call":
  1. The teacher simply shows how to find the dual of a given graph without providing any explanation.
  2. While the teacher is creating the dual, the students write down what steps they think are necessary.
  3. Using a new graph, the teacher finds the dual by following instructions given by students.
  4. Summary and clarification: The teacher summarizes the steps taken, giving precise terminology—students revise the steps they created in step 2.
• "Dueling Duals": Using the board, pairs of students practice creating duals by having a "duel" by creating the dual the fastest.
• EXTENSION: Have the students find the dual of a dual (they will find a graph isomorphic to the original).

Closure
• Students write at least two paragraphs of different applications (not discussed in class) that can be modeled using graphs.

VI. Relation to NCTM Goals

1. Valuing Mathematics—This unit helps students recognize that math is more than just numbers. A "picture" is often more efficient and equally powerful to numbers.

2. Confidence in Math Ability—As in the previous unit, "What's Your Type" fosters a sense of accomplishment for the students.

3. Becoming Mathematical Problem Solvers—The students must create models for given applications.

4. Communicating Mathematically—In this unit, students are communicating via their graphs and their paragraphs.

5. Reasoning Mathematically—Students are encouraged to look for patterns which delineate different types of graphs. Plus, in their writing, they must extend their ideas to think of their own types of applications suited for graph models.
I. Lesson Identification

GRADE: 8

TOPIC: Juice It Up!

II. General Objectives

Nature & Scope
This unit will develop connections between a graph model and the mathematics used to describe the model—specifically, the mathematics used are incidence matrices. Working with matrices serves as a first introduction to difference equations, which in turn will lead to calculus.

Major Outcomes
1. Students describe an application in words, with a graph, and with numbers.
2. Students appreciate the elegance of matrices.

III. Specific Objectives

1.1 Students describe the current flow of a simple circuit.
1.2 Students construct a graph model of a circuit.
1.3 Students develop the incidence matrix corresponding to the graph.
2.1 Students create a diagram relating the node incidence matrix of a circuit with the brightness of light bulbs (essentially students are relating a node incidence matrix with the current flow of the circuit).
IV. Rationale

No single model is appropriate for all situations. Graphs are versatile and efficient; they become unwieldy, however, when too complex. Mathematics—numbers—can preserve the essential relationships of a graph, such as number of vertices, degree of each vertex, number of edges, and types of graph while dispensing with inefficient models. Thus, matrices extend the usefulness of graphs to model complex applications.

V. Content

Set
- “Predict This!”—Students brainstorm several famous predictions (i.e. Nostradamus; World Series winners) and Pros & Cons of predictions (i.e. making incorrect predictions).

Body
- Present “Rationale.”
- Define NODE INCIDENCE MATRICES
  1. Uses of node incidence matrices.
  2. Specific application: electrical circuits.
     - Define electrical circuits
     - Describe flow of current —> high voltage to low voltage
- Students view several electrical circuits and make predictions (please see the attached sketches of circuits):
  1. Equal battery sources for each circuit.
  2. Equal light bulbs.
  3. Equal wires.
  4. PREDICT: Which light bulbs will burn brighter?
- Students construct node incidence matrices and a graph for each circuit.
- Students test predictions.
  Why didn’t predictions match up?
  - Discuss current flow through networks—resistance of the edges and how they are connected determines the brightness of the lights.

Closure
- Students create a diagram relating an incidence matrix with the brightness of the bulbs. In other words, will knowing the node incidence matrix enable people to make predictions on the brightness of the light bulbs? (Knowing how the circuit is connected they can determine and describe the current flow through the circuit.)

VI. Relation to NCTM Goal

1. Valuing Mathematics—This unit helps students learn the value of “numbers” in the sense that matrices handle unwieldy graphs.
2. **Confidence in Math Ability**—Achievement in this unit comes in successfully completing a circuit and then describing the circuit using mathematics and words.

3. **Becoming Mathematical Problem Solvers**—Students are approached with the problem of how to accommodate complex graphs. Also, they must create a working circuit. Solving these problems teaches the students new skills and boosts confidence.

4. **Communicating Mathematically**—This unit allows students to communicate their model in three different ways. They must write about it, construct a graph model, and develop the corresponding incidence matrix.

5. **Reasoning Mathematically**—Making the connection between “numbers” and the graph is an important step in a student’s thinking. Developing both models, simultaneously, allows for a true understanding of the connection by the students.
CIRCUITS TO USE FOR MAKING PREDICTIONS

Figure 4.1. A Simple Circuit.
Figure 4.2. A Circuit in Series.

Figure 4.3. A Circuit in Parallel.

Figure 4.4. A Circuit in Parallel/Series Combination

Figure 4.5. A Circuit in Series/Parallel Combination
I. Lesson Identification

GRADE: 10

TOPIC: You Take the High Road, I’ll Take the Low Road

II. General Objectives

Nature & Scope
This unit focuses on the characteristics and properties inherent in a graph to determine paths and circuits. The students are learning to recognize patterns and then use the patterns described in theorems to choose paths and circuits based on the needs of the application.

Major Outcomes
1. Students differentiate between a path and a circuit.
2. Students identify and describe an Eulerian path and circuit.
3. Students identify and describe a Hamiltonian path and circuit.

III. Specific Objectives

1.1 Students write, in their own words, the definition of a path and a circuit.
2.1 Students list the requirements of a graph to possess an Eulerian path and circuit.
2.2 Students construct an Eulerian path and circuit on a given graph.
3.1 Students list the requirements of a graph to possess a Hamiltonian path and circuit.
3.2 Students construct a Hamiltonian path and circuit on a given graph.

IV. Rationale

Everything is in motion, especially people moving from one place to another. Depending on the application, there are certain mathematical patterns to satisfy the necessary motion. This unit uses theorems and patterns to determine particular motions.
V. Content

Day 1
Set
- "What's the Connection?" Students describe the connection between the following examples:
  1. A baseball player and a homing pigeon—both complete circuits.
  2. Red Riding Hood and a package—both complete a path.
  3. Golfers and Students—both complete circuits.

Body
- Present "Rationale."
- Define PATH and CIRCUIT.
- Guided Discovery—students work in pairs.
  1. Students are given several different graphs and are to find paths and circuits according to certain qualifications. Sometimes the path and circuit must contain every edge and sometimes the qualification is to contain every vertex.
  2. While determining the paths and circuits, the students are to note which types of graphs have paths and circuits according to the given qualifications.
  3. The pairs present their findings.

Closure
- Students write the reasons why some graphs have paths and circuits while some do not.

Day 2
Set
- Students write two examples of paths and circuits they travel in their lives, daily.

Body
- Review "Rationale."
- Review types of paths and circuits—those containing every edge or every vertex.
- Define EULERIAN PATH and CIRCUIT and HAMILTONIAN PATH and CIRCUIT.
- Differentiate Eulerian and Hamiltonian paths and circuits, relating the definitions to yesterday's findings.
- Students practice finding paths and circuits:
  1. Yes or No, does the graph contain: an Eulerian path or circuit?
     Hamiltonian path or circuit?
  2. If yes, highlight.

Closure
- Write four paragraphs describing applications of Eulerian paths and circuits and Hamiltonian paths and circuits.
VI. Relation to NCTM Goal

1. **Valuing Mathematics**—This unit fosters the idea that the qualities inherent to a model determine what sort of patterns—paths and circuits—are present.

2. **Confidence in Math Ability**—This unit is exciting for students because the students discover the patterns that dictate what type of paths and circuits are present in a graph. The students achieve the information themselves.

3. **Becoming Mathematical Problem Solvers**—In discovering the patterns of graphs containing different types of paths and circuits, students not only become more confident in their math ability, they are also becoming mathematical problem solvers.

4. **Communicating Mathematically**—In this unit, the students must describe the characteristics of graphs which allow for paths and circuits and also show the actual path and circuit.

5. **Reasoning Mathematically**—In discovering the paths and circuits, the students are looking for patterns. Finding patterns helps to simplify problems. Once problems are identified and simplified, students are able to work on the solution. Identifying patterns is an important mathematical reasoning skill.
I. Lesson Identification

GRADE: 2nd year in college

TOPIC: You Make the Call

II. General Objectives

Nature & Scope
In this unit, students use the principles of graph theory and networks to solve problems current on their campus. Using algorithms and specific graph terminology, the students will apply their skills in the contexts of an assignment problem, a shortest path, and a maximum flow problem.

Major Outcomes
1. Students define an assignment problem and apply the principles to making an assignment.
2. Students define a shortest path and apply the principles to finding a shortest path.
3. Students define a maximum flow problem and use the principles to find the maximum flow of a network.
4. Students appreciate the use of algorithms and their corresponding computation time.

III. Specific Objectives

1.1 Students write, in their own words, the definition of an assignment application.
1.2 Students list three types of applications requiring the principles of an assignment application.
1.3 Using an application of interest to the student, related to the college campus, the students make assignments, including a write-up defending their model.
2.1 Students write, in their own words, the definition of a shortest path application—including the definitions of both Prim and Kruskal.
2.2 Students list six types of applications requiring the principles of a shortest path application—three for Prim and three for Kruskal.
2.3 Using an application of interest to the student, related to the college campus, the students find the shortest path (using either Prim or Kruskal, whichever is appropriate), including a write-up defending their shortest path.

3.1 Students write, in their own words, the definition of a maximum flow network application.

3.2 Students list three types of applications requiring the principles of a maximum flow application.

3.3 Using an application of interest to the student, related to the college campus, the students finds the maximum flow of a network, including a write-up defending their network flow.

4.1 Students write a one-page paper describing the relationship of computer computation time with algorithms.

IV. Rationale

Our world includes many discrete systems. Handling these systems requires specific models. Graph theory and networks are the most appropriate for modeling discrete systems and making decisions based on the context of the system. Specific types of graphs model each system and proven theorems decide what course of action is appropriate and best for each system.

V. Content

Day 1

Set
- “Dinner Party”—The class will be attending a formal dinner and must arrange dates for the evening.
   - Everyone first writes their top three choices.
   - Free-for-all! See how the class solves the problem.
   - Students present matchings and defend process for finding the “dates.”

Body
- Present “Rationale.”
- Define ASSIGNMENT APPLICATIONS—include the following terms: bipartite graph, complete matching, cover.
- The Teacher models how to make a matching assignment using preferences determined in the beginning of class—for purposes of the demonstration, limit the sets to contain four of each sex—compare the efficiency of the algorithm to the processes used during the “Set” activity.
- Students, individually, solve five assignment problems.

Closure
- Students write three examples of assignment applications found on a college campus.

Day 2

Set
• “Choose a direction”—given a map of the U.S. displaying several routes with distances from Portland, Oregon to Dallas, Texas, in groups of three, students determine the shortest route. Then the groups present their findings, defending the process used to find the shortest route.

Body
• Review “Rationale.”
• Define SHORTEST PATH/MINIMUM COST SPANNING TREES—include the following terms: tree, weighted edge, Prim’s Algorithm, Kruskal’s Algorithm.
• The Teacher models how to find the shortest path of the example used during the set using both Prim and Kruskal’s Algorithm—compare these algorithms to the ideas generated by the “Set” activity.
• Students, individually, solve five shortest path problems.

Closure
• Students write three examples of shortest path/minimum cost spanning tree applications found on a college campus.

Day 3
Set
• “Let it Flow”—Given a system of pipes, the flow of water into the system and the diameter of each pipe in the system, students working in groups of four find the maximum flow of water out of the system. The groups must defend their process and solutions.

Body
• Review “Rationale.”
• Define MAXIMUM FLOW NETWORKS—include the following terms: network, weighted edge, source, sink, minimum cut, and Dijkstra’s Algorithm.
• The Teacher models how to find the maximum flow of the example used during the set using Dijkstra’s Algorithm—compare the maximum flow found with the algorithm to that found during the “Set” activity.
• Students, individually, solve five maximum flow problems.

Closure
• Students write three examples of maximum flow applications found on a college campus.

Day 4
Set
• “Game Show”—Via question/answer, students review the past three days of activities. Include “prizes” for correct answers.

Body
• Review “Rationale.”
• Brainstorm—Using ideas generated by the students at the end of each day, list types of discrete applications of networks found on college campuses.
• Class divides into pair. Each pair choosing an application to model.

Closure
• Students describe model and provide a “best guess” of how the system will be solved.

Day 5
Set
• “Pop Quiz”—Students must describe the three types of applications practiced in this unit. Answers must include steps of the algorithms, or process, used to solve each type of application.

Body
• Review “Rationale.”
• Pairs present the findings of their systems, including visual of their modeled system before applying an algorithm, plus an example of the system during the application of the algorithm, and the final optimal system.

Closure
• Students have an opportunity to “add” anything to their pop quizzes.
• Students write a one-page paper explaining the relationship of the studied algorithms to computer computation time.

VI. Relation to NCTM Goals

1. Valuing Mathematics—This unit fosters the idea of algorithms as a tool to solve discrete systems. Graph theory fosters an appreciation for algorithms because even simple systems are time-consuming to solve by hand. Yet the beauty of these models is that the process is the same whether working on a system with five nodes or 500. Learning to value the process, rather than the numbers involved, is a big step in a student’s developing mathematical literacy.

2. Confidence in Math Ability—This unit relies on giving students an opportunity to solve the applications themselves. Although they may generate inefficient processes, they make an attempt that does work. To reiterate, this fosters a sense of achievement, which in turn fosters a sense of confidence.

3. Becoming Mathematical Problem Solvers—This unit allows students the first attempt to solve the presented problems. Also, the students identify problems present in their college campus community and use these contexts as problems to solve. The major project of modeling systems found in their community is important because it is the students, themselves, that identify the problems and create solutions.

4. Communicating Mathematically—Students communicate in three ways throughout this unit. They write, orally explain, and construct graph models. Communicating orally, in the form of defending their solutions to their applications, creates an opportunity for the students to communicate their own ideas, rather than simply repeating what the teacher tells them to think about a certain problem.

5. Reasoning Mathematically—Much of this unit relies on the students for generating not only problems, but solutions. These activities challenge their reasoning skills and provide practice for honing them.
The purposes of this thesis are twofold. To explore graph theory and to demonstrate how to teach this mathematical concept using a spiral curriculum model. In reviewing the previous chapters, these goals were met. This thesis provides an accurate and reliable (if somewhat short and simple) mathematical reference to graph theory, and demonstrates how to teach graph theory by providing a working plan of implementing these mathematical concepts via five modules that spiral through an educational experience, beginning in kindergarten and ending in college.

Any of the sources on graph theory cited in this thesis provide excellent guides to study this topic further. For me, graph theory is an area of interest that I anticipate pursuing further in graduate school. I hope the reader will leave with a sense of excitement and appreciation for graph theory, similar to mine.

I also hope that readers have a realistic sense of how a spiral curriculum works. A spiral curriculum is difficult to implement because of the time necessary to coordinate a plan that encompasses kindergarten through high school graduates. But this should not be a deterrent from using the ideas inherent to a spiral curriculum. The idea of revisiting certain themes, with increasing sophistication, is valuable. The value of the spiral curriculum is that it is logical and just makes plain sense to connect and reiterate concepts throughout the entire career of the student.
REFERENCES


