ABSTRACT

In 1960 Claude Berge introduced the concept of perfect graphs. Since then, perfect graphs have been a very important and rewarding aspect of graph theory. There are many classes of perfect graphs and many properties that make a graph perfect. The Weak Perfect Graph Theorem and the Strong Perfect Graph Theorem are two results that were conjectures for many years until as recently as 1979 and 2002, respectively. In this paper we will discuss some of classes of perfect graphs, the intuition behind the proofs surrounding the concepts of perfect graphs, and a couple of applications to perfect graphs.

1. INTRODUCTION

A graph $G$ is an ordered pair $(V(G), E(G))$ consisting of a nonempty set of vertices, $V(G)$, and a disjoint set, $E(G)$, of edges. For example, the graph $G$ in Figure 1 has $V(G) = \{2,3,4,5\}$ and $E(G) = \{a,b,c,d,e,f,g,h,i,j\}$.

![Figure 1](image1.png)

An edge $e$ of a graph $G$ is said to be incident with a vertex $v$ if it starts or ends at $v$. Two edges that share a vertex are also incident. In Figure 1, edge $c$ is incident to vertices 1 and 4; likewise, vertices 1 and 4 are each incident to edge $c$. Two vertices (1 and 4) which are incident to a common edge ($c$) are said to be adjacent and two edges ($c$ and $a$) which are incident to a common vertex (1) are also adjacent. An edge that is incident to the same vertex twice is called a loop. If two edges are incident to the same two vertices, these edges are parallel edges. Figure 1 shows an example of a loop, edge $i$, and a set of parallel edges, $g$ and $j$. A simple graph is one that contains no loops or parallel edges, thus the graph in Figure 1 is not simple. (For more information on basic graph theory, see [1]).

There is more than one way to draw a single graph—the locations of the vertices and how the edges connect them are irrelevant as long as the same vertices and edges are present and incident to the correct vertices and edges. An example of this is shown in Figure 2. The two graphs look different at first sight, but on further inspection they are
actually the same, or, they are isomorphic—the edges and vertices are just depicted differently. A planar graph is one which can be drawn on the plane with all of its edges intersecting only at their ends. In other words, a planar graph can be drawn with no two edges intersecting each other. Thus, Figure 2 depicts a planar graph.

![Figure 2](image1.png)

A graph $H$ is a subgraph of a graph $G$ if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. In Figure 2, the graph containing vertices 1, 2, and 3 and edges $f$ and $a$ is a subgraph. A spanning subgraph of $G$ is a graph $H$ where $V(G) = V(H)$. Now, suppose that $V'$ is a nonempty subset of $V$. The subgraph of $G$ induced by $V'$ is the graph whose vertex set is $V'$ and whose edge set $E''$ is all the edges in $G$ which have both ends in $V'$. In mathematical terms, $E'' \equiv \{(i,j) \in E : i,j \in V'\}$. Similarly, let $E'$ be a nonempty subset of $E$. Then the subgraph of $G$ induced by $E'$ is the graph whose edge set is $E'$ and whose vertex set $V''$ is the set of vertices which have an end of an edge in $E'$ touching them. Again, in mathematical terms, $V'' \equiv \{i,j \in V : (i,j) \in E'\}$. (For more information on subgraphs, see [1] and [7].)

![Figure 3](image2.png)

A walk in a graph $G$ is an alternating finite sequence of vertices and edges $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$, where each $v_i \in V$ for $i = 0, 1, 2, \ldots, k$ and each $e_i \equiv (v_{i-1}, v_i) \in E$ for $i = 1, 2, \ldots, k$. A trail from vertex $v_0$ to vertex $v_k$ is a walk from vertex $v_0$ to vertex $v_k$ which contains no repeated edges. A path from vertex $v_0$ to vertex $v_k$ is a trail from vertex $v_0$ to vertex $v_k$ which contains no repeated vertices. A cycle is a trail from vertex $v_0$ to vertex $v_0$ which has no repeated vertices other than vertex $v_0$. Sometimes, the term cycle is used to denote a graph corresponding to a cycle. A $k$-cycle is a cycle of length $k$ and is odd or even depending on whether $k$ is odd or even. A chord
is an edge that connects two vertices which are not adjacent in a cycle. In graph $G$ of Figure 3, edges $c$ and $d$ are chords. In addition, $G$ is considered to be connected if there is a path from vertex $u$ to vertex $v$ for every pair of vertices $u, v \in V$. (For more on structures of graphs, see [1] and [7].)

2. CURRENT WORK

From now on, we will only discuss simple, finite graphs. Before getting to the definition of perfect graph, there are some more terms that need to be defined. The complement $G'$ of a simple graph $G$ is the simple graph with its vertex set consisting of all vertices in $G$ and edge set consisting of all the edges not in $G$. In other words, two vertices are adjacent in $G'$ if and only if they are not adjacent in $G$. A stable set of $G$ is a subset of vertices $v^*$ on which the subgraph induced by $v^*$ contains no adjacent vertices. The cardinality of the largest stable set of $G$ is denoted $\alpha(G)$. The opposite of a stable set of $G$ is a clique, which is a graph in which all pairs of vertices are adjacent. The maximum clique number of a graph $G$ is the cardinality of a largest subgraph of $G$ which is a clique, and is denoted by $\omega(G)$. In Figure 3, the graph $G$ contains cliques of sizes four, three, or two and $\omega(G)$ is four. The four vertices that make up the maximum size clique are 1,2,3,4. A hole is a subgraph that is induced by a chordless cycle containing at least four vertices. An odd hole is a hole that contains an odd number of vertices. It is important to note that a chordless cycle containing three vertices is not considered a hole. Graph $G$ in Figure 3 also contains an odd hole of size five and the vertices making up the hole are 2,4,5,6,7. (For more on graph theory definitions, see [1].)

The concept of vertex coloring is important when studying perfect graphs; therefore it will be helpful to discuss the main points of vertex coloring. A $k$-coloring of a graph $G$ is an assignment of at most $k$ colors, $1,2,3,\ldots,k$, to each of the vertices of $G$ in which no two distinct adjacent vertices have the same color. The chromatic number, $\chi(G)$, of $G$ is the minimum $k$ for which $G$ is $k$-colorable, or the minimum number of colors it takes to color $G$ in a way that no two adjacent vertices are the same color. (For more on vertex coloring see [1].)

Now, to the important definition: “A graph $G$ is perfect if $G$ and each of its induced subgraphs, $G'$, has the property that its chromatic number $\chi(G')$ equals the size of a maximum clique $\omega(G')$” [6]. This definition alone doesn’t sound so bad, but when one starts dealing with graphs of huge sizes, inspecting each induced subgraph of a graph is computationally inefficient. Therefore we need more tractable ways to characterize perfect graphs.

3. MOTIVATION TO STUDYING PERFECT GRAPHS

To some, the theory of perfect graphs and the proofs and results surrounding perfect graphs are quite interesting in their own right, but these graphs also have applications to real world problems, which are of great interest to other mathematicians. Some of these applications are, but not limited to, the vertex coloring problem, various optimization problems including the vehicle routing problem, and applications involving linear programming. Each of these applications is useful in its own way, but we will only have time to briefly discuss two of them.
3.1

Alan Tucker [8] investigated the following problem, which was posed to the Urban Science Department at Stony Brook by the city of New York. There is a set of sites $S_i$ which must be serviced $k_i$ times a week ($1 \leq k_i \leq 6$; in this case the visit was to pick up garbage). One wishes to find a minimal or near-minimal set of truck tours for a week such that each site is visited $k_i$ times and these tours can be partitioned among six days of the week in a way so that no cite is visited more than one time per day. In general, this is an extremely difficult problem and the method proposed for attacking it is a modification of an algorithm due to Clarke and Wright [2]. The algorithm starts out with an inefficient set of tours and successively tries to improve the set of tours. Also, this method only gives a near-minimal set of tours. In further studying perfect graphs, my hope is to be able to sufficiently explain the solution of this problem with the help of Tucker’s paper.

3.2

Another interesting application to perfect graphs is the role they play in integer linear programming (ILP) problems. Before discussing the actual role of perfects graphs, it is necessary to give an introduction to the area of linear programming. A linear programming problem (LP) can be defined as an optimization problem with a linear objective function and with linear equalities or inequalities as constraints. For instance,

$$\text{maximize } x_1 + x_2 \text{ such that}$$

$$x_1 + 2x_2 \leq 4$$
$$4x_1 + 2x_2 \leq 12$$
$$-x_1 + x_2 \leq 1$$
$$x_1 \geq 0, x_2 \geq 0$$

where $x_1$ and $x_2$ are the variables, is an LP. In this example, there are two knowns and five constraints, all of which are linear inequalities. The first two constraints, $x_1 \geq 0$ and $x_2 \geq 0$, are called nonnegativity constraints, the other constraints are the main constraints and the function to be maximized is the objective function. This problem can be solved by graphing the set of points in the plane that satisfies all the constraints and then finding the point that maximizes the value of the objective function. This is easily done since there are only two variables. The problem could have many variables and be subject to many constraints. (For more on Linear Programming see [4].)

By and large, linear programming problems do not yield integer optimal solutions when they are solved. However, in a network or discrete optimization problem such as the “multiple depot vehicle routing and resource allocation” (VR) problem described in Whitty [9], optimal solutions with fractional values are meaningless. Thus, an LP model for the problem is not sufficient; we need to model the problem as an ILP. Since ILP’s are known to be NP-hard, meaning there is no known efficient time algorithm to solve ILP to optimality and there probably never will be, it is useful to know under what conditions the VR problem could be easily solved. Chvatal [5] proved that if the underlying network, or graph, of VR is perfect then the associated LP yields at least one
integer optimal solution. An illustration of VR is shown in Figure 4 where the vertices 1 through 6 on graph R in Figure 4 depict six depots from which we wish to dispatch goods every day by railroad. We want to pick the best-stocked depots but are subject to the constraint that we choose only one depot per network clique so head-on collisions are avoided. If this graph is perfect, then it is possible to find an integer solution to the VR problem.

![Figure 4](image)

**4. FUTURE PLANS**

In studying perfect graphs on a deeper level, I plan to present some of the intuition behind the proofs of the theorems regarding perfect graphs. I also plan to study the different subclasses of graphs that are perfect and the properties that make a graph perfect. In order to illustrate these properties, I plan to construct a few of my own perfect graphs as well as a few graphs that are not perfect. I will also continue working with the applications of perfect graphs that I have presented here in order to illustrate the motivation to studying the area of perfect graphs. In list form next semester I plan to:

- Study the properties that make a graph perfect
- Study the subclasses of graphs that are perfect
- Construct a few perfect graphs to illustrate properties
- Construct a few graphs that are not perfect and show why they are not
- Present some intuition behind some of the proofs involved with perfect graphs
- Continue to work on the two applications I have already started looking at
5. REFERENCES


