NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

EXAMINING A SUBPROBLEM OF THE FREQUENCY ASSIGNMENT PROBLEM USING A CONFLICT GRAPH

by

Donald W. Hintze

March 1990

Thesis Advisor: Kim A.S. Hefner

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S)

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION
Naval Postgraduate School

6b OFFICE SYMBOL *(If applicable)*
Code 53

7a NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

6c ADDRESS *(City, State, and ZIP Code)*
Monterey, California 93943-5000

7b ADDRESS *(City, State, and ZIP Code)*
Monterey, California 93943-5000

8a NAME OF FUNDING / SPONSORING ORGANIZATION

8b OFFICE SYMBOL *(If applicable)*

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS *(City, State, and ZIP Code)*

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE *(Include Security Classification)*
EXAMINING A SUBPROBLEM OF THE FREQUENCY ASSIGNMENT PROBLEM USING A CONFLICT GRAPH

12 PERSONAL AUTHOR(S)
Hintze, Donald W.

13a TYPE OF REPORT
Master's Thesis

13b TIME COVERED
FROM ________ TO ________

14 DATE OF REPORT *(Year, Month, Day)*
1990, March

15 PAGE COUNT
89

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Frequency Assignment; Conflict Graph; Graph Theory

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

There are many problems associated with communication networks. One of the more familiar ones is the frequency assignment problem. Many approaches and techniques have been used in the past in an attempt to solve this problem. This thesis examines a subproblem of the frequency assignment problem, which aids the decision-maker in placing additional links in a network, once a frequency assignment is found. Given a conflict graph for a communications network, the problem involves finding the maximum number of arcs in the corresponding digraph. This digraph is a worst case model for the actual network and will show which additional links may be added to the network in order to enhance communication capabilities. An algorithm was developed to help solve this problem after lower and upper bounds were established for its optimal solution. The algorithm obtains a solution which falls within the bounds and achieves the bounds in special cases.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL
Prof. Kim A.S. Hefner

22b TELEPHONE *(Include Area Code)*
(408) 646-2198

22c OFFICE SYMBOL
Code MA/Hk

UNCLASSIFIED

Approved for public release; distribution is unlimited

Examining a Subproblem of the Frequency Assignment Problem Using a Conflict Graph

by

Donald W. Hintze
Captain, United States Marine Corps
B.S., United States Naval Academy, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
March 1990

Author: Donald W. Hintze
Donald W. Hintze

Approved by: Kim A.S. Hefner
Kim A.S. Hefner, Thesis Advisor

for John Thornton, H. Fredricksen
John R. Thornton, Second Reader

Harold M. Fredricksen
Harold M. Fredricksen, Chairman
Department of Mathematics

ABSTRACT

There are many problems associated with communication networks. One of the more familiar ones is the frequency assignment problem. Many approaches and techniques have been used in the past in an attempt to solve this problem. This thesis examines a subproblem of the frequency assignment problem, which aids the decision-maker in placing additional links in a network, once a frequency assignment is found. Given a conflict graph for a communications network, the problem involves finding the maximum number of arcs in the corresponding digraph. This digraph is a worst case model for the actual network and will show which additional links may be added to the network in order to enhance communication capabilities. An algorithm was developed to help solve this problem after lower and upper bounds were established for its optimal solution. The algorithm obtains a solution which falls within the bounds and achieves the bounds in special cases.

Keywords: radio frequency communications. (KR)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

The term "graph" used in this paper denotes something quite different from the graphs that one may be familiar with from calculus or analytic geometry. The graphs used here are geometrical figures made up of points called nodes or vertices and lines connecting these points called arcs or edges. Because graphs are a simple and systematic way to represent a binary relation among many systems, graphs are a very useful tool in modeling. Some of the systems that have been modeled using graphs arise from engineering, environmental, social science and economic problems.

The theory of graphs has been around for many years, dating back to the middle of the 18^{th} century. In 1736, the famous Swiss mathematician, Leonhard Euler (1707-1783), used graph theory to solve the Konigsberg Bridge problem, which has become famous since then. The city of Konigsberg is located on the banks and on two islands of the Pregel River. At that time, the various parts of the city were connected by seven bridges as shown in Figure 1.1.

The people of Konigsberg wanted to know if it was possible to start at one point, cross each of the seven bridges exactly once, and return to the starting point. Euler showed that it was not possible, and his methods laid the foundations of

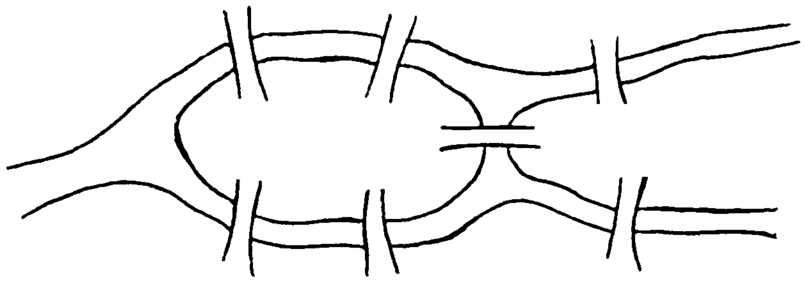


Figure 1.1 The Konigsberg Bridge Problem

graph theory. From that point on, graph theory has always had close ties to applications.

Some of the applications which have been studied using graph theoretic principles include food webs, transportation problems, map colorings, communication networks and many more. The one application that will be looked at in this paper is a communication networking problem which consists of many different parts or submodels. Before giving the background of the communication problem, it should be known that this paper was written with many definitions, notations, and terms that are fairly standardized within the field of graph theory but may not be familiar to the reader.

General definitions or terms that are used in the chapters to follow will be found at the beginning of each chapter so as to enhance the continuity of the material presented. The terms and definitions from the field of graph theory that are used in the paper are adopted from the well-known graph theorist, Fred S. Roberts [Ref. 1], unless otherwise stated.

They are generally accepted by all graph theorists. Definitions and terms that are more specific to the material being presented will remain with that section.

II. NATURE OF THE PROBLEM

The purpose of this chapter is to present background material on the communication problem, to summarize recent work that has been completed on several communications problems which use different techniques, and to state the specific objectives and approach of this thesis.

A. DEFINITIONS/TERMINOLOGY

As mentioned earlier in the introduction, a node is simply a point which graphically represents an object of a given set. Other terms for a node are point and vertex. An edge is a line which is drawn between two nodes and represents a relation between the two. Two nodes or vertices with an edge between them are said to be adjacent. The relation may also be directed from one vertex to another, by drawing an arrow in that direction on the edge. This is called an arc. Other terms used for an arc are directed edge or directed link. To be more precise it is necessary to understand the basic difference between a graph and a directed graph or digraph. A digraph D = (V,A) is a pair where V is the set of vertices and A is the set of arcs, (u,v), where u and v are elements of V. The notation V(D) and A(D) correspond to the vertex set and the arc set of digraph D, respectively. A graph G = (V,E) is a pair where V is a set of vertices and E is a set of edges, (w,x), where w and x are elements of V. Again, V(G) and E(G) are the vertex and edge set of graph G. Two things to conclude from these definitions are: (1) that arcs go with digraphs and edges go with graphs; and (2) an arc is an ordered pair of vertices in a digraph which represents a direction and an edge is a pair of vertices which are not ordered in a graph.

There are a couple of communication terms that must also be defined. The word <u>net</u> is a communication term used for a collection of stations that are linked together by a common frequency. <u>Frequency assignment</u> is a process of assigning frequencies to the transmitters in a communication network in such a way as to meet certain objectives.

The final concept to be discussed in this section will be the concept of computational complexity. Because there are many variables and factors associated with a communications network, there are many problems and subproblems involved. Many of these smaller problems, even taken separately, do not possess efficient solutions. One class of these problems which are very difficult to solve is the class of <u>NP-complete</u> problems. Aho, Hopcroft and Ullmann claim there are two approaches to this type of problem,

> When such a problem is encountered, it is often useful to determine if the inputs to the problem have special characteristics that could be exploited in trying to devise a solution, or if an easily found approximate solution could be used in place of the difficult-to-compute exact solution. [Ref. 2:p. 306]

The approximate solution that they talk about in the second approach is normally obtained from a <u>heuristic</u>, which is an algorithm that quickly produces good, but not necessarily optimal solutions. Many believe that no algorithm, which solves an NP-complete problem is substantially more efficient than the obvious approach of trying all possibilities. This process is referred to as enumeration.

Another way to describe NP-complete problems has to do with the time complexity involved in finding a solution. Time complexity is generally described as being either average case or worst case, the latter usually being easier to compute. Worst case time complexity describes the largest amount of time required by an algorithm to solve the given problem, which is dependent upon the characteristics of the input. An algorithm is said to be polynomial-bounded if its worst case time complexity is bounded by a polynomial function of the input size. Thus, a problem is normally considered NP-complete when there is no polynomial-bounded algorithm to solve the problem. The list of NP-complete problems has grown extensively in the last decade, and many have important practical applications. Garey and Johnson [Ref. 3] have accumulated a large collection of NP-complete problems and describe them briefly in their book on NP-completeness.

B. PROBLEM DESCRIPTION

A radio communications network, or system, consists of radio stations that are equipped with transmitters and receivers. When a station in the network receives information from another station on the net, a link is said to exist from the transmitting station to the receiving station. The interconnection of the stations and links can be viewed as a set of nodes representing the radio stations, joined together by arcs representing the links that exist in the network. Communication networks may consist of stations in which every radio at a station is a receiver/transmitter. So, there will always be the same number of receivers as transmitters in those networks. Others will assume that there are separate transmitters and separate receivers located at each station. In those networks, there may not be the same number of transmitters as receivers.

The radio frequency communication environment has become increasingly complex as more and more receivers and transmitters have been added to perform certain tasks in both military and civilian areas. Therefore, the decision-makers, such as frequency managers and communications officers, face many problems in managing a communications system. This growth in the overall use of the electromagnetic spectrum "has necessitated a careful design and management of any communications network to allow for accurate, fast, interference-free, and reliable communication systems." [Ref. 4:p. 829] The objectives mentioned here are the familiar requirements associated with any effective communication system. Some systems may also require security, particularly when talking about military applications.

There are obviously many different problems associated with communication systems, but the one that will be examined in detail in this paper is the frequency assignment problem.

> One of the most critical design problems in a radio communication network is the assignment of transmit frequencies to stations (nodes) so that designated key communication links will not be jammed due to self interference. [Ref. 5:p. 133]

Thus, the overall objective in assigning frequencies to the network is to achieve the most reliable system possible. There are many forms of the frequency assignment problem that arise in managing a system. However, most of the frequency assignment problems are actually optimization problems with the following form: Given a number of transmitters in a network that require frequencies, those frequencies are to be assigned, subject to certain constraints that minimize different types of interference or minimize the amount of spectrum utilized. Another form of the frequency assignment problem occurs when additional nets must be added to an existing communication network. In this case, a frequency must be assigned to the new net so that the constraints are still satisfied and the interference remains minimal.

The frequency assignment process is obviously highly dependent upon many factors found within the communications network. Some of these factors include: (1) number of stations in the network; (2) location of the stations; (3) radio equipment used at the stations; (4) links to be established among the stations; (5) frequencies available for assignment; and (6) climatological conditions present at the network. It is not hard to understand how these factors could affect frequency assignment. Some of these factors are not as easy to control as others, and some may be given as initial conditions to the problem. For instance, the climatological conditions and equipment available may be factors which cannot be altered. Alternately, the number of stations, locations, equipment, etc., may already be determined and the problem is to find an assignment for the given setup.

Hale [Ref. 6:p. 1497] wrote that the first frequency assignment problem occurred when it was discovered that interference was present among several transmitters assigned to the same or closely-related frequencies. So, the first frequency problem was to minimize or eliminate this type of interference among transmitters in a network. This is only one type of interference that may be present in a system. Other types will be discussed later. In addition, the constraints needed to even partially solve the problems will be addressed.

The solution that was first used to solve this problem was to assign different noninterfering frequencies to each of the transmitters if possible. This method wasted a lot of the frequency spectrum, and with the growing demands on the spectrum, was not considered an acceptable solution. Therefore, the managers of the spectrum had to consider different approaches to the frequency assignment problem.

One particularly interesting fact about the frequency assignment problem is that many are equivalent to generalized graph coloring problems as Hale [Ref. 6:p. 1497] points out. This approach will be further investigated later in this paper. At that point, the graph coloring problem will be defined in greater detail. Graph coloring problems are optimization problems like the majority of frequency assignment problems. These problems are also some of the most famous optimization problems that have ever been studied.

It is important to understand that the frequency assignment problem was not always thought of as an optimization problem as it is today. In fact, the problem was so diverse and complex that formal mathematical models were not even considered as a way to approach the problem until the 1960's. However, the interest in formal frequency assignment models has increased significantly. This can be seen from the number of articles appearing on the subject. This thesis will also examine the problem using a formal model using graph theory principles. But despite all of the findings and conclusions about the problem, many frequency managers and planners still do not believe that formal models are a viable approach to many varieties of frequency problems that occur in the real world. [Ref. 6:p. 1497]

A major reason for this skepticism is that many believe that the formal models can handle only a limited number of the constraints or variables that are associated with a real world problem. This reason is easily justified by the various constraints which different authors choose to model when researching the problem. For instance, Zoellner and Beall [Ref. 7] consider only cochannel constraints while concluding that their method obtains significant spectrum savings. This leads to a discussion of the different constraints associated with a communications network.

Zoellner and Beall [Ref. 7:p. 314] classify the electromagnetic compatibility constraints found in communications networks into three groups:

1. cochannel constraints
2. adjacent channel constraints
3. cosite constraints.

The following descriptions of the three groups are from Mathur, Salkin, Nishimura, and Morito [Ref. 4:pp. 829-830]. Cochannel constraints arise when certain pairs of transmitting nets located in the same area and with sufficient power so that their transmission areas overlap, are assigned the same frequency or channel. Thus two different nets may exist at two different stations altogether or even within the same station. In the latter case, the station has multiple transmitter/receiver radios. The second type of constraint, adjacent channel constraints, exists when certain pairs of frequencies must be separated by a minimum amount in order to avoid interference. For example, two transmitters cannot use adjacent frequencies. This separation could be measured in terms of a certain percentage of the transmitting frequencies or as a specific interval of the spectrum. For example, a necessary separation of 3 kHz could be imposed. Finally, <u>cosite</u> constraints describe conditions in which subsets of transmission frequencies should not be assigned to nets which are in close proximity to each other in order to avoid potential interference.

<u>Intermodulation</u> occurs when secondary frequencies are created from an interaction of multiple frequencies that are transmitting simultaneously in the network. When these secondary frequencies interfere with other frequencies in the network, intermodulation interference occurs. So intermodulation is a type of cosite constraint which is hard to satisfy in the frequency assignment problem.

Mathematically, cochannel and adjacent channel constraints are relatively easy to describe. Therefore, systems with these conditions have been studied and investigated in some detail. Many different techniques and approaches have been used in the past to consider a network of this type including: a graph coloring model, set-partitioning procedures, nonlinear programming techniques, and integer programming models. Some of these particular methods will be described later. The first model considered in greater detail uses integer programming techniques for a system with adjacent channel constraints and cosite constraints. The cosite constraints in the model are of the intermodulation type.

Once the different techniques are summarized, this thesis will describe the formulation and analysis of an algorithm that was created by the author. The algorithm was developed to solve a particular subproblem of a communications network that was discovered while studying the frequency assignment problem. This subproblem is closely associated with the frequency assignment problem and helps the decision-maker to make additional decisions once the optimal assignment is found.

C. MODELING THE FREQUENCY ASSIGNMENT PROBLEM AS AN INTEGER PROGRAM

1. Introduction

This section describes the approach and techniques used by Mathur and others [Ref. 4] to model the frequency assignment problem. Their approach was basically to model the communication network using an integer programming model with the associated adjacent channel and cosite constraints.

2. Definitions

Z is the notation used for the set of integers {...-3,-2,-1,0,1,2,3...}. The following definition of a linear program is taken from Bazaraa and Jarvis [Ref. 8:p. 2]. A linear programming problem (linear program) is a problem to minimize or maximize a linear function, called an objective function, in the presence of linear constraints of the inequality and/or the equality type. The following is an example of a linear programming problem.

Minimize	$c_1x_1 + c_2x_2 + c_3x_3$	(objective function)
subject to	$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \le b_1$	(constraint 1)
	$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \le b_2$	(constraint 2)

The coefficients c_1, c_2 and c_3 and the a_{ij} are known coefficients and x_1, x_2 and x_3 are decision variables to be determined. The two constants b_1 and b_2 are known constants which represent requirements to be satisfied, and are minimal requirements in this example. A set of variables $\{x_1, x_2, x_3\}$ which satisfy all the constraints is called a feasible solution to the linear program. One of the best feasible solutions to the program is called an optimal solution. In this example, a feasible solution that gives the minimal value for the objective function would be an optimal solution. (A linear program can have more than one optimal solution.) The value of the objective function corresponding to an optimal solution is called the optimal value of the linear program.

If any component of the objective function or constraints contains nonlinear functions, then the problem is called a nonlinear programming problem (nonlinear program). An integer linear programming problem, or integer program, is a linear programming problem wherein some or all of the decision variables are restricted to be integer-valued. There are also integer nonlinear programming problems. Both are called integer programs for brevity.

3. Problem Formulation

The communication network under consideration consists of a station that has a set of N transmitting frequencies $\{F_1, F_2, \ldots, F_N\}$ and one receiving frequency, F_R. Without loss of generality, the frequencies can be thought of as being positive integers. As mentioned earlier, intermodulation occurs when secondary frequencies are created from the transmitting frequencies. The secondary frequencies, called <u>intermodulation frequencies</u>, can only occur as integer linear combinations of the transmitting frequencies [Ref. 4:p. 831]. Intermodulation interference occurs when one of the intermodulation frequencies matches one of the receiving frequencies. Therefore, in mathematical terms, intermodulation interference occurs in the network whenever

$$F_R = \sum_{i=1}^{N} X_i F_i \qquad X_i \in Z \qquad (2.1)$$

The sum $Q = \sum_{i=1}^{N} |X_i|$ is called the <u>order of interference</u> and determines the level of interference expected in the network. If Q is small, then interference will be strong. So part of the problem in finding a frequency assignment, when cosite constraints are modeled, includes minimizing this order of interference. The integer programming model described thus far is:

Model 1

$$\textbf{Minimize} \qquad Q = \sum_{i=1}^{N} |X_i| \tag{2.2}$$

$$\textbf{subject to} \qquad \sum_{i=1}^{N} X_i F_i = F_R \qquad X_i \in Z \tag{2.3}$$

Once this nonlinear integer program is solved, which is an NP-complete problem, it can be determined whether or not the current set of frequencies are acceptable based upon the optimal value of Q. We call this optimal value Q*. If Q* is less than a specified, previously determined, tolerance level, then this assignment is unacceptable. This means that the transmitting frequency, F_R, can be interfered with by some combination of the transmitting frequencies, and the frequency manager would have to change the assignment in some way. After the change, the new assignment would have to be analyzed using the same model in order to calculate the new value of Q*. The overall process of selecting a frequency assignment is actually a problem of iteratively finding an acceptable assignment of the entire network.

In reality, Model 1 does not accurately describe the constraints which are found in a real-world communication network. The linear integer combination of transmitting frequencies does not have to match the receiving frequency exactly before intermodulation interference occurs. The combination has only to fall within a certain neighborhood around the receiving frequency to cause interference. So constraint 2.3 in Model 1 should be modified to look like

$$F_R - GB \le \sum_{i=1}^{N} X_i F_i \le F_R + GB \qquad (2.4)$$

where GB is a known parameter called the guard band. In other words, if the combination $\sum_{i=1}^{N} X_i F_i$ falls within the interval $[F_R - GB, F_R + GB]$, then the given assignment will interfere with F_R. Thus the model changes to:

Model 2

Minimize $\quad Q = \sum_{i=1}^{N} |X_i|$

subject to $\quad F_R - GB \le \sum_{i=1}^{N} X_i F_i \le F_R + GB \qquad X_i \in Z$

This model can be considered a worst case approach because it assumes that all N of the station's transmitting frequencies are active simultaneously and this may never be the case. In real world networks, it is not very probable that all N nets at a station will be simultaneously active. Therefore, another parameter, K, ($K \le N$) may be defined as the maximum number of nets that are simultaneously active. In this case, when the model is solved, intermodulation interference must only be checked for integer combinations of K or fewer frequencies. However, it is possible that all nets could be simultaneously active, in which case $K = N$. Mathematically, this becomes another constraint in the model which Mathur and others [Ref. 4:p. 831] derived.

This approach can, in fact, help the decision-maker in assigning frequencies to the network, by evaluating the order of interference prior to the actual assignment. Mathur and others describe frequency assignment as the problem which,

> ...requires that the decision maker assign to each net of the communication network a frequency from a set of resource frequencies (say) $F_1, F_2, \ldots, F_N$ such that for each ship, the assigned frequencies do not create intermodulation interference of order less than an acceptable order. [Ref. 4:p. 832]

This description obviously is directed towards a naval application since it mentions ships. However, the word "station" could be substituted for "ship" in order to apply to any communication network.

In addition to this interference constraint, if adjacent channel constraints (<u>separation constraints</u>) are to be considered, then the frequencies at each station must be separated by a minimum amount. This separation minimum is often defined as a percentage (α%) of the transmitting frequency. For example, given two frequencies F_i and F_j in a network, F_i would not cause any separation problem with F_j unless F_i fell within the interval $[F_j - \alpha\%(F_j), F_j + \alpha\%(F_j)]$.

The frequency assignment problem which considers intermodulation interference and separation constraints can be solved using a technique called branch and search, which is outlined by Mathur and others [Ref. 4:pp. 835-836]. In the algorithm, start by assigning frequencies to two nets in the system and test the pair with respect to the intermodulation interference and separation limits that are given for the network. This is accomplished in several steps.

(1) <u>For each pair of nets</u>, test to ensure frequencies are minimally separated using the α parameter given.

(2) <u>For each station</u>, test to ensure frequencies do not create intermodulation interference that is not acceptable using the order of interference, Q, that is prescribed.

If steps (1) and (2) are satisfied, then the process is continued by assigning resource frequencies to those nets in the network that still require frequencies, and then this assignment is tested. If the current assignment fails to satisfy both (1) and (2), then it must be modified and the modification retested. The assignment is modified by the change of one or more of the frequencies.

Mathur and others [Ref. 4:p. 834] applied the model to a specific network consisting of five ships and six nets, with frequencies ranging from 6750-29004 kHz. The parameters used for the problem were: guard band (GB) = 6 kHz, separation parameter (α) = 5, lowest acceptable order (Q) = 6, and K = 2. This gives some idea of the magnitude of the values which the parameters may take on.

Of the two constraints considered in the problem, the separation constraint is the easier to evaluate. Since this constraint requires that two frequencies be separated by a certain amount, the decision-maker can usually meet this constraint by quick inspection if the network is small. Whereas, the intermodulation constraint may be difficult to evaluate even for small networks. This is due to the nature of the integer linear combination required by the constraint. For example, consider the frequency assignment problem with three frequencies given in Table 2.1. It is easy to see that

TABLE 2.1

FREQUENCY ASSIGNMENT EXAMPLE

Frequency 1	6,750 kHz
Frequency 2	6,500 kHz
Frequency 3	12,755 kHz

parameters: GB = 6 kHz $\alpha = 5$ Q = 6 K = 2

Frequencies 1 and 2 do not meet the separation constraint because they are so close together. But it is not as apparent that an integer combination of Frequency 1 and Frequency 2, $(3F_2-F_1)$, is within the guard band around Frequency 3, and would cause intermodulation.

4. Observations/Conclusions

The network under consideration assumes several conditions that may not be present in a realistic communication network. It assumes that there is only one receiving frequency at each station and that each station is not in close proximity to another station in the network.

If there were more than one receiving frequency at a station, then it would be possible for the transmitting frequencies to combine and interfere with any of the receiving frequencies. In this case, the integer program would have to be solved as many times as there are receiving frequencies at the station. For example, if there are five receiving frequencies and five transmitting frequencies, (N = 5), at a station, then the model would have to be solved five times, with constraint 2.4 changing each time to cover the five receiving frequencies. This must be done for each station in the network, so the approach is obviously very time-consuming for large N.

If two stations are closely located in the network, then another factor must be considered. In this case, it is possible for transmitting frequencies at the two stations to combine and interfere with any receiving frequency at either station. For instance, consider a network with two stations, Station A with N transmitting frequencies and N receiving frequencies, and Station B with P transmitting frequencies and P receiving frequencies. When solving the integer programming model for this network, the index i must range over all N+P transmitting frequencies each time the program is solved. The program must also be solved N+P times because there are that many receiving frequencies.

The search procedure eventually terminates with either an assignment that meets all interference constraints or with the conclusion that there is no interference-free assignment, based upon the given resource frequencies. This second conclusion would not be acceptable in most cases, since the communication network must still function in order to carry out its mission. At that point, the frequency manager would have two courses of action to consider. He could decide to use the frequency assignment which gave the least amount of interference, or he could request additional frequencies to work with and thus increase the number of resource frequencies available to him.

There are several conclusions to be drawn about the overall branch and search method. Each step in the method involves solving an integer program which is an NP-complete problem in itself and is thus very time-consuming for a large network. If K is small, the time required to solve the integer programs, and so the overall problem, is relatively minimal as concluded by Mathur and others [Ref. 4]. They claim that two iterations of the process required about ten seconds of CPU time using a DEC-20 computer for a network made up of ten stations and 20 nets. However, in a more realistic network of about 100 stations and 200 nets, the time required to solve the problem may be unacceptable. This would be especially true if the parameter K was larger than two.

Because the solution of both the overall branch and search method and the integer program is an enumerative process, the results cannot be guaranteed in a timely fashion.

(In the future, this method may be implemented with quicker results by taking advantage of parallel processors.) As concluded by Mathur and others,

> ...it is very hard to predict, a priori, if given a particular communication network and frequency resource list, whether the algorithm for the underlying frequency assignment problem will converge in a reasonable amount of computer time. [Ref. 4:p. 838]

So other approaches should be considered to solve the frequency assignment problem, before any real use of the model can be implemented successfully.

The next section will describe a method that considers the frequency assignment problem as a graph coloring problem.

D. LOOKING AT THE FREQUENCY ASSIGNMENT PROBLEM AS A GRAPH COLORING PROBLEM

1. Introduction

Since different forms of the frequency assignment problem are similar to the graph coloring problem of graph theory, this enables a whole new approach to the problem. Zoellner and Beall [Ref. 7:p. 314] wrote that B.H. Metzger was the first to recognize that the classical graph coloring problem was equivalent to the frequency assignment problem where only cochannel constraints are modeled. The vertices represent the stations in the network and the edges represent the cochannel restrictions between pairs of stations.

Because of the similarity between the two problems, if new methods are found to solve the graph coloring problem, these same techniques can be applied in solving the frequency assignment problem. This was the idea used by Cameron [Ref. 9] when he looked at the graph coloring problem as a sequence of set-covering problems. His approach is discussed in this section but there are several definitions and terms to be explained first.

2. Definitions

The following definition of the generalized graph coloring problem is taken from Roberts [Ref. 10:pp. 97-98]. Given a graph G = (V,E) with n vertices, the goal is to assign a color to each vertex in G, in such a way that if two vertices are adjacent, they receive a different color. If the assignment can be performed using k colors, the assignment is called a k-coloring of G, and G is said to be k-colorable. Obviously, any graph with n vertices can be colored with n colors, but this is not as interesting as finding the minimum number of colors required. Finding the smallest number of colors is the generalized graph coloring problem. The smallest number k, for which the graph G is k-colorable, is called the chromatic number of G and is denoted $\chi(G)$. For example, the graph in Figure 2.1 has chromatic number 3, because vertices 1, 2 and 7 can be assigned color #1, vertices 3, 5 and 6 can be assigned color #2, and vertex 4, color #3. This coloring is not unique for the graph in Figure 2.1, but a chromatic number less than three is not possible. This can be easily proven since the graph contains triangles made up of vertices 1, 3 and 4 and vertices 1, 4 and 5.

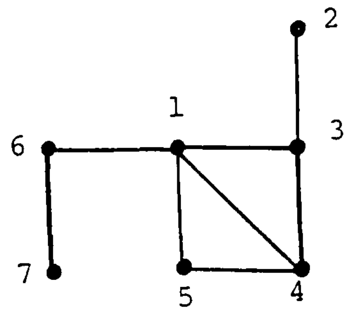


Figure 2.1 A Coloring Problem

For certain classes of graphs, the chromatic number is known. Other classes may have lower and/or upper bounds on the chromatic number. However, determining $\chi(G)$ for an arbitrary graph is a difficult task and is in fact an NP-complete problem.

Terms and definitions associated with the set-covering problem are taken from the article by Cameron [Ref. 9]. Consider a rectangular <u>binary</u> matrix A with elements $a_{ij} \epsilon (0,1)$ where a_{ij} is the element in the i^{th} row and j^{th} column. The j^{th} column of the matrix is said to "cover" every row for which $a_{ij} = 1$. A cover for the <u>matrix</u> is the collection of columns which together cover every row in the matrix. A collection of the smallest number of columns that are required to cover A is called a <u>minimum cover</u> for matrix A. Thus, the problem of finding a minimum cover for a binary matrix is called a <u>set-covering</u> problem. Figure 2.2 is an example to help illustrate this concept. An important fact to remember is that the minimum cover does not have to be unique, as it is for matrix A in Figure 2.2.

$$A = \begin{matrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{matrix}$$

Problem: Find a minimum cover for matrix A.

Column 1 covers Rows 1 and 3.
Column 2 covers Rows 1 and 5.
Column 3 covers Rows 2, 3 and 5.
Column 4 covers Rows 1, 4 and 5.
Column 5 covers Row 4.
Column 6 covers Rows 3 and 5.

Columns 1, 3 and 4 cover matrix A.
Columns 2, 3 and 5 cover matrix A.

Solution. Columns 3 and 4 are a minimum cover for A.

Figure 2.2 A Set-Covering Problem

3. Problem Formulation

The problem investigated here is actually to find the chromatic number of a graph, which models a communication network, using the set-covering approach [Ref. 9]. Consider a communication network consisting of n stations and m links which is modeled by a graph, G, with n vertices and m edges. A p-coloring of G is the objective of the method where p is less than n and greater than one. If p = 1, then there would be no assignment possible, given that at least one edge exists in G ($m \neq 0$). If $p \geq n$, this would be trivial to solve, since there are only n vertices to be colored.

Thus a formulation of the problem is described which contains a hypothesis to be tested. This hypothesis is that a given graph has a chromatic number less than or equal to an integer p [Ref. 9:p. 320]. The smallest integer p that satisfies the hypothesis formulated is the chromatic number of the graph.

The method proceeds by building the unique binary matrix, for the communication system modeled, so that certain requirements are met. For instance, the first step in building this matrix consists of building a binary submatrix of size (mp) rows and (np) columns. (The details on constructing the matrix can be found in Reference 9.) When a minimum cover for the overall matrix is generated, this submatrix ensures that no adjacent vertices will be given the same color. The next step adds (np) more columns and n rows to the existing matrix to ensure that each vertex is assigned at least one color. Finally, the current matrix is augmented by (np) more rows. A one in position (i,j) implies that either vertex i is assigned color j while a zero in position (i,j) implies that vertex i is not assigned color j when the minimum cover is generated. In other words, the minimum cover must imply a selection and cannot be neutral in the coloring of a node.

It is important to understand that the matrix generated by the above steps is a binary matrix. When the different submatrices are constructed, the requirements also determine where the 0's and 1's occur in the matrix.

Consider the network depicted in Figure 2.3 which consists of $n = 5$ nodes and $m = 6$ links. If the hypothesis assumed the integer p was 3, then the binary matrix would look like that of Figure 2.4. The location of the 1's in the matrix will not be discussed, but their location is obviously important in selecting the minimum cover for the matrix.

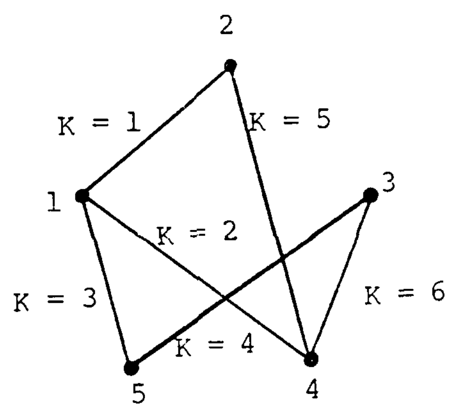


Figure 2.3 Example Network

It is pointed out, however, that there should be 3np+2mp total 1's in the matrix which contains n+np+mp rows and 2np columns. Therefore, a communication system with $n = 5$ nodes, $m = 6$ links, and a hypothesis of $p = 4$ colors would possess a binary matrix of size 49 X 40 and contain 108 1's.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1			1																										
2		1			1																									
3			1			1																								
4	1									1																				
5		1									1																			
6			1									1																		
7	1												1																	
8		1												1																
9			1												1															
10							1						1																	
11								1						1																
12									1						1															
13				1						1																				
14					1						1																			
15						1						1																		
16							1			1																				
17								1			1																			
18									1			1																		
19																1	1	1												
20																			1	1	1									
21																						1	1	1						
22																									1	1	1			
23																												1	1	1
24	1															1														
25		1															1													
26			1															1												
27				1															1											
28					1															1										
29						1															1									
30							1															1								
31								1															1							
32									1															1						
33										1															1					
34											1															1				
35												1															1			
36													1															1		
37														1															1	
38															1															1

Figure 2.4 Binary Matrix for Network in Figure 2.3

Thus, even a moderate-sized communication network would generate a very large binary matrix. For example, a network with n = 10 nodes, m = 25 links, and p = 5 colors would generate a binary matrix of size 185 X 100.

As can be seen, the formulation of the problem depends directly on the hypothesis to be tested, that is, the integer p. If the graph is indeed p-colorable, then there exists a minimum cover for the matrix which consists of exactly np columns. If the graph is not p-colorable, then the minimum cover will contain more than np columns. [Ref. 9:p. 321]

4. Observations/Conclusions

The technique employed above to find a minimum cover determines the minimum number of channels required for a frequency assignment problem with cochannel constraints. As Cameron [Ref. 9:p. 321] points out in his evaluation of the approach, the technique can be formulated and solved for moderate-size problems, but larger networks may be very time-consuming.

The set-covering problem is a specific case of the generalized covering problem taught in many introductory linear programming courses. Linear programming techniques have been applied with large success in solving these covering problems. There exists commercial software available today which solves many classes of linear programming problems and most are based on implementation of the Simplex Algorithm. One such software package is LINDO, a flexible and powerful linear program system that is available for both personal and mainframe computers.

With the speed of today's computers, the technique employed by Cameron could be used to find a solution in a reasonable amount of time. The solution would only be limited by the size of the matrix the software could handle. Again, this may be a factor when considering a communication network with many stations and links, because the binary matrix would be very large, yet it is always very sparse.

Another point to consider is that this technique would probably be used in conjunction with other theorems and techniques that would provide upper and lower bounds on the value of p. In this way, the number of different values for p may be decreased or increased in searching for the minimum value.

Again, this technique is used to find the minimum number of frequencies necessary to equip a network with cochannel constraints only. Thus, this approach would not be helpful if the communication system were modeled with adjacent channel or cosite constraints.

The procedure described in this section was applied to a frequency assignment problem with cochannel constraints, but is actually just another technique in solving the generalized graph coloring problem. There exist other algorithms which are fairly efficient in arriving at an answer to the graph coloring problem. One of the most familiar of these algorithms is the Welsh and Powell algorithm described by Dossey, Otto, Spence, and Eynden [Ref. 11:p. 115]. This algorithm does not solve the problem since it does not always give an optimal solution. However, it does give a coloring in good time.

Once again, it is important to realize the nature of the overall problem. Both the graph coloring problem and the set-covering problem are NP-complete problems, and are therefore hard to solve quickly when seeking an optimum solution.

The next section will describe a technique which solves the frequency assignment problem using a nonlinear programming model. This is the third and final technique to be discussed in this chapter, before discussing the subproblem associated with the frequency assignment problem.

E. SOLVING THE FREQUENCY ASSIGNMENT PROBLEM WITH A NONLINEAR PROGRAMMING MODEL

1. Introduction

This section will describe the mathematical programming model formulated by Allen, Helgason, and Kennington [Ref. 5], to assign frequencies to stations in a communications network. Their model considers both cochannel constraints and adjacent channel constraints, and is actually a specific type of nonlinear program.

2. Problem Formulation

The authors assume that the communications network under consideration consists of N radio stations that each have one transmitter and several receivers. The transmitter is assigned its own frequency while the receivers are tuned to the transmitting frequencies of other station transmitters. A channel is associated with each transmitting frequency the way frequencies are assigned channels in a television set. [Ref. 5:pp. 133-134]

Once again, the frequency assignment problem takes on the general form of an optimization problem with objective function and constraints. The objective is to assign one of F transmit channels to each transmitter while minimizing any self-interference caused by cochannel and adjacent channel conditions. A link in the network is considered to be interfered with if either of the two following situations occur: (1) a station receives a signal on a given channel while a neighboring station transmits on an adjacent channel; or (2) a station receives two signals on the same channel that are less than a certain parameter apart in signal strength [Ref. 5:p. 134].

Allen and others [Ref. 5:p. 134] used the following notation in describing their mathematical model. Let $f \quad \{1,2,...,F\}$ be a channel and $n \quad \{1,2,...,N\}$ be a station. Let $x_{fn} = 1$, if channel f is assigned to station n and 0 otherwise; $g(x_1,x_2,...,x_F)$ is a weighted number of links that possesses interference given the assignment $(x_1,x_2,...,x_F)$. Therefore, the mathematical model derived by Allen and others [Ref. 5:p. 134] is as follows.

Minimize	$g(x_1, x_2, \ldots, x_F)$		
Subject to	$x_{fn} = 1$	for all n	(2.5)
	$x_{fn} \in (0,1)$	for all f,n.	(2.6)

The objective function seeks to minimize interference in an assignment while constraint 2.5 above ensures that each station receives exactly one frequency or channel. Constraint 2.6 implies that x_{fn} is a binary variable as described above.

The model given above falls into a class of nonlinear programs called <u>binary nonconvex nonlinear</u> programs. Allen and others use a special application of the convex simplex algorithm to obtain an optimum solution for the model [Ref. 5:p. 134]. Further details of the model and the method of solution will not be described in this thesis. It is not the intent to discuss here the convex simplex algorithm that was used or the computer programming methods developed to solve the nonlinear program. The purpose here is to emphasize some of the results and conclusions that can be drawn from the overall approach used by Allen and others.

3. <u>Observations/Conclusions</u>

Specialized software was written to solve the mathematical model and was tested on five versions of a real world communication network that consisted of 43 stations and 63 links. The model was implemented on a CDC Cyber-875 computer with good results. All five runs were completed in less than one minute of CPU time and the final results were quite close to the optimum [Ref. 5:p. 139]. An optimum solution, in this case, is one in which the frequency assignment would generate no self-interference at all within the network.

Allen, Helgason, and Kennington conclude their article by claiming that their optimization model and software can greatly assist the decision-maker "in obtaining near-optimal solutions for the frequency assignment problem." [Ref. 5:p. 139] They also claim their model can handle very large scale communication networks.

F. SUMMARY

This chapter has described several methods that have been used in the past to solve certain frequency assignment problems. The nature of the problem suggests some optimization method is the best way to approach it. The methods previously described use optimization techniques including integer programming, nonlinear programming, set-covering and graph coloring. These methods do not always produce optimal solutions, and when they do, they are often too time-consuming to be of practical use. That is why it is so important to increase the effectiveness of a network, without changing the frequency assignment, once it is found. This is the subproblem examined in this thesis.

The next chapter will provide additional background about frequency assignment problems in general, and then discuss the details leading to a solution of the subproblem.

III. MAXIMIZING A DIGRAPH GIVEN ITS CONFLICT GRAPH

A. INTRODUCTION

This chapter can be considered the heart of this thesis and will lead to an algorithm designed by the author to solve a subproblem of the frequency assignment problem. First, two different approaches in modeling the frequency assignment problem will be discussed. One of these approaches will provide the basis for the method used by the author. The next section will be used to introduce and explain terms and definitions that will be required later in the chapter. After that, additional background information will be presented before discussing some of the assumptions made in the author's model.

Once the problem is set up, the next section will describe the algorithm used to solve the subproblem. This algorithm helps the decision-maker to create a more efficient communications network, without changing the number of frequencies assigned to the network. After the algorithm is analyzed, it will be used on a sample network in order to illustrate the results. Finally, the chapter ends with a summary and conclusions section.

B. MODELING THE FREQUENCY ASSIGNMENT PROBLEM

The general frequency assignment problem discussed in this thesis is an optimization problem. The objective is to minimize self-interference while assigning frequencies to receivers in the network. As mentioned before, the frequency assignment problem is equivalent to a graph coloring problem. Thus, to model a network using the principles of graph theory is very logical. In the several papers written on the frequency assignment problem, there seems to be two general methods to model the communication network as a graph coloring problem.

The first method is the more general and does not model specific constraints such as adjacent channel, cochannel, and cosite constraints. This method is described as follows. Given a communication network that consists of transmitting and receiving stations, model the network by a digraph where each station is a vertex. The arc (x,y) in the digraph implies that station x can communicate directly to station y. If there are two different stations, a and b, that can communicate directly to a third station, c, then there is a possible conflict when a and b both try to communicate with c simultaneously. This approach to the communications problem is where the concept of a conflict graph first arose. A conflict graph is actually a type of competition graph which is discussed later in this chapter. The solution to this problem involves finding a coloring for the conflict graph which determines a feasible frequency assignment.

The second method, to model a frequency assignment problem as a graph coloring problem, goes into more detail about the different constraints put on the system. For example, Baybars [Ref. 12:pp. 257-262] uses this approach to model the frequency assignment problem that has both cochannel and adjacent channel constraints. Baybars models this type of communication network with a graph that has two different types of edges in it. The vertex set denotes the stations as before. One type of edge models the cochannel restrictions between two stations and is represented by a solid line. The second type of edge, a dashed line between two vertices, indicates that an adjacent channel restriction exists between the two stations.

Figure 3.1 is an example of a graph for this type of model. Once again, the solution to this problem involves coloring the graph described above. Since this graph coloring includes additional requirements not found in a generalized graph coloring, Baybars [Ref. 12:p. 259] calls it a constrained graph coloring.

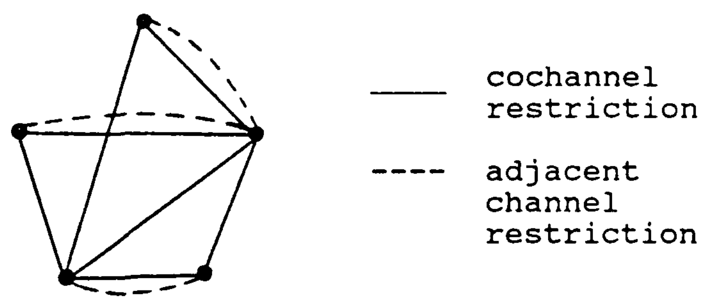


Figure 3.1 Network Model

In coloring a graph, the set of vertices receiving the same color is called a <u>color class</u>. Each color class is associated with a frequency in the model. Because the adjacent channel restrictions imply that two stations cannot receive adjacent channels, the color classes must possess an adjacency quality in order to model this additional constraint. Therefore, let C_j, $j = 1,2,\ldots,k$ denote the j^{th} color class and call two classes C_j and C_{j+1} adjacent. In this way, by coloring the graph so that: (1) no two vertices with cochannel restrictions are assigned to the same color class; and (2) no two vertices with adjacent channel restrictions are assigned to adjacent color classes, the coloring admits an assignment to the problem. Baybars [Ref. 12] solves this problem using an integer programming model, which finds an optimal solution for the coloring. However, the computer time required in solving the model is very large even for small networks.

The first method described, which leads to a conflict graph for a network, is the approach used to arrive at the algorithm to be discussed in this chapter. Before discussing the nature of the problem, there are many definitions, terms, and notations that will be explained in the next section.

C. DEFINITIONS/TERMINOLOGY

The idea of a graph and a digraph were discussed earlier in Chapter I. Now specific attributes of the two will be explained further. A graph is <u>connected</u> if there exists a path between every pair of vertices in the graph. In other words, the graph has "one" piece or <u>component</u>. In Figure 3.2, graphs G_1 and G_2 are connected, but G_3 and G_4 are not. A

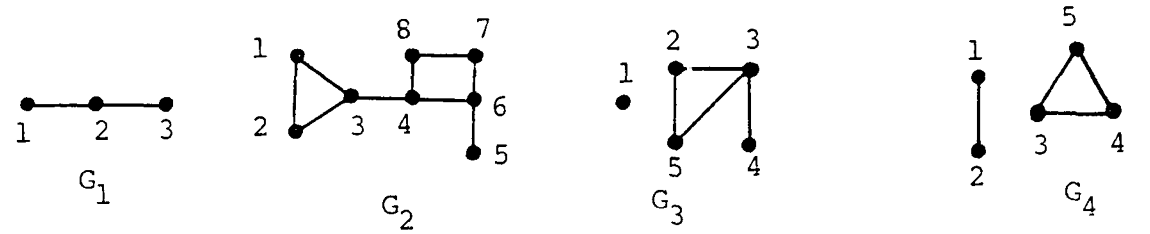


Figure 3.2 Graphs

<u>complete graph</u>, K_n, is a graph with n vertices in which every pair of distinct vertices is joined by an edge. Figure 3.3 shows complete graphs, K_n, for n = 2-5. A <u>directed complete</u>

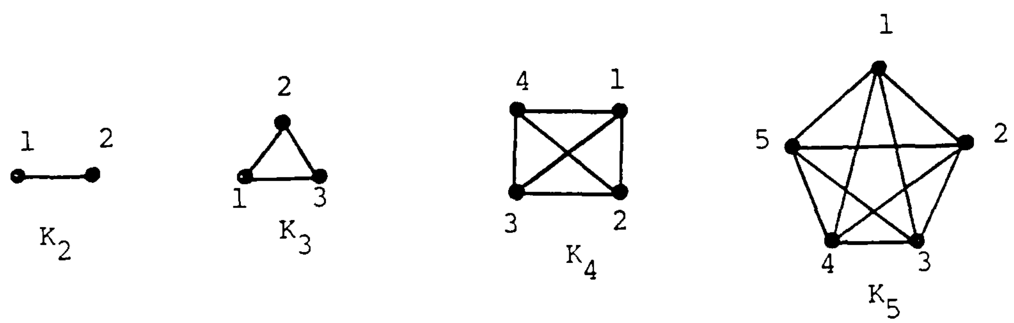


Figure 3.3 Complete Graphs

<u>graph</u>, DK_n, is a graph with n vertices in which there exists an arc (x,y) and (y,x) for every distinct pair of vertices x and y in V(D), as in Figure 3.4. If G = (V,E) is a graph, a

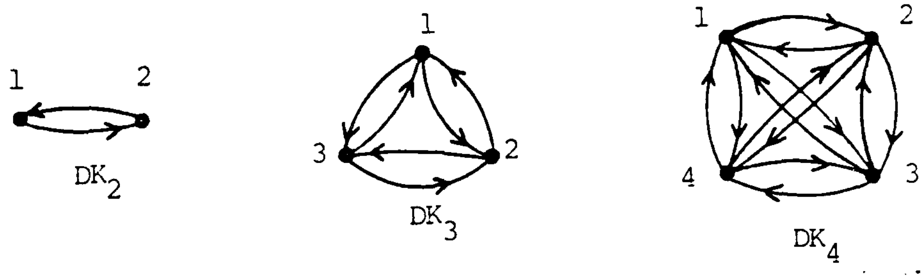


Figure 3.4 Directed Complete Graphs

subgraph of G is a graph H = (W,F) where W is a subset of V and F is a subset of the edge set E. Any complete subgraph K of a graph G, is called a clique of G. If K has k vertices we call it a k-clique or a clique of order k. If k = 1, then k is just a vertex. This is called a trivial clique. When k = 2, the clique is an edge in the graph, and when k = 3, the clique is referred to as a triangle. A clique is maximal if it is not contained in a larger clique. The algorithm discussed later in this chapter uses only maximal cliques. An edge clique covering (ECC) is a collection of cliques which covers all the edges of G. That is, the union of all cliques in the ECC is E(G). A minimal edge clique covering of G is one whose cardinality is least among all edge clique coverings of G. The clique covering number of G, denoted cc(G), is the cardinality of its minimum clique covering. Note here that a graph may have more than one minimum clique covering. Cardinality of a set S will be denoted as |S|, and is simply the number of elements in set S.

A cycle of length k is a path in a digraph, $v_1,e_1,v_2,e_2,\ldots,v_k,e_k,v_{k+1}$, where $k > 0$, $v_1 = v_{k+1}$, and all of the vertices $v_1,v_2,\ldots,v_k$ and, hence, all of the edges $e_1,e_2,\ldots,e_k$, are distinct [Ref. 11:p. 97]. Digraphs which do not have any cycles are called acyclic digraphs. It is generally said that a digraph which is not acyclic "has cycles," since the term cyclic is not used in this case.

The next concept to be explained is a combinatorial problem which involves matching items, subject to certain restrictions. This matching problem leads to the concept of a system of distinct representatives (SDR). Dossey and others [Ref. 11:pp. 235-237] explain the idea of an SDR in the following manner. Let $S_1,S_2,\ldots,S_n$ be a finite sequence of sets, where the S_i are not necessarily distinct. An SDR for $S_1,S_2,\ldots,S_n$ is a sequence $x_1,x_2,\ldots,x_n$ such that $x_i \in S_i$ for $i = 1$ to n, and such that the elements x_i are all distinct. The finite sequence of sets, S_i, are usually grouped into a collection, **F**, called a family of sets. For instance, consider the family of sets $\mathbf{F}_1$, where $\mathbf{F}_1 = \{S_1,S_2,S_3,S_4\}$ and

$$S_1 = \{1,2,3\}$$
$$S_2 = \{1,3\}$$
$$S_3 = \{1,3\}$$
$$S_4 = \{3,4,5\}.$$

Then an SDR for $\mathbf{F}_1$ would be 2,1,3,4. This SDR is not unique for $\mathbf{F}_1$. It is also possible that **F** does not have an SDR, as shown below. Suppose $\mathbf{F}_2 = \{S_1,S_2,S_3,S_4\}$ where

$S_1 = \{2,3\}$
$S_2 = \{2,3,4,5\}$
$S_3 = \{2\}$
$S_4 = \{3\}$.

In this case, $\mathbf{F}_2$ has no SDR because S_1, S_3 and S_4 have only two distinct elements among their union, and it would be impossible to choose three distinct elements for these sets. This leads to the famous theorem which was discovered by Philip Hall. Hall's Theorem explains the conditions which are necessary in order for a sequence of sets, **F**, to have an SDR.

Hall's Theorem: The family $\mathbf{F} = \{S_1,S_2,\ldots,S_n\}$ has an SDR if and only if whenever I is a subset of $\{1,2,\ldots,n\}$, then the union of the sets S_i for $i \in I$ contains at least as many elements as the set I does.

Hall's Theorem can best be understood by an example from Dossey and others [Ref. 11:p. 237]. Hall's Theorem will be used to show that $\mathbf{F}_3$ has an SDR. Let $\mathbf{F}_3 = \{S_1,S_2,S_3\}$ where

$S_1 = \{1,3,5\}$
$S_2 = \{1,2\}$
$S_3 = \{2,5\}$.

The subsets I of $\{1,2,3\}$ and the corresponding union of sets S_i are as shown in Table 3.1. By Hall's Theorem, since every set on the right has at least as many elements as the corresponding set on the left, then $\mathbf{F}_3$ has an SDR. One SDR for $\mathbf{F}_3$ is 3,1,2. The concept of an SDR proves to be a major part of the algorithm to be described.

TABLE 3.1

SDR

I	Union of Sets, S_i, $i \in I$
0	0
{1}	{1,3,5}
{2}	{1,2}
{3}	{2,5}
{1,2}	{1,2,3,5}
{1,3}	{1,2,3,5}
{2,3}	{1,2,5}
{1,2,3}	{1,2,3,5}

The last idea to be covered in this section is the competition graph. The notion of a competition graph was first introduced by Cohen [Ref. 13] in 1968 during his study of food web models in ecology. The competition graph of a digraph $D = (V,A)$ is the graph $G = (V,E)$ where $V(G) = V(D)$, and $(x,y) \in E(G)$ if and only if $x \neq y \in V(D)$ and (x,z) and $(y,z) \in A(D)$ for some $z \in V(D)$. The notation $C(D)$ will be used to indicate the competition graph corresponding to digraph D. The competition graph described above is actually a specific case of the <u>generalized competition graph</u>, $G(D,B,C)$, described below.

Suppose $D = (V,A)$ is a digraph and B and C are sets of vertices in D. $G(D,B,C)$ is an undirected graph with vertex set equal to B and with an edge between two distinct vertices x and y of B if and only if for some a in C, (x,a) and (y,a) are elements of $A(D)$. Therefore, a competition graph is just a generalized competition graph where $B = C = V(D)$. The generalized competition graph was first introduced by Roberts and has applications to many problems.

Competition graphs, along with several generalizations and extensions, have been studied extensively in the last ten years. Some of these extensions include confusion graphs, row graphs, niche overlap graphs, and conflict graphs. The conflict graph introduced earlier is a case of the generalized competition graph. In this case, B is the set of transmitting stations and C is the set of receiving stations found in the network. Define digraph D ith vertex set equal to $B \cup C$. An arc (x,y) in the digraph implies that station x in B is transmitting directly to station y in C. Then G(D,B,C) is the conflict graph for the transmitting stations. This means that two stations in the network could possibly communicate simultaneously to a third receiving station and thus a conflict would arise. Because this competition graph is actually a conflict graph for the transmitting stations, the coloring for this graph determines the frequencies assigned to the transmitters.

Another term associated with the competition graph is the competition number. The competition number of a graph G, denoted k(G), is the smallest integer k such that $G \cup I_k$ is a competition graph of some acyclic digraph. Here I_k is the notation used to describe a graph of k isolated vertices and no edges. Calculating the competition number for an arbitrary graph is not a simple task. In fact, the problem of computing the competition number for a graph is another example of an NP-complete problem.

Note that the competition number of a graph is only defined for graphs which are competition graphs of acyclic digraphs. This initial restriction to acyclic digraphs can be attributed to Cohen. Cohen observed that most food web models were acyclic, disallowing forms of cannibalism, and thus the restriction to this type of digraph was introduced. This leads into the next section which will cover some of the history behind competition graphs.

D. BACKGROUND

Many people went on to study competition graphs after Cohen introduced the idea in 1968. In his studies, Cohen was primarily interested in the case where D was an acyclic digraph. Others who have written papers on the acyclic case include Dutton and Brigham [Ref. 14], Lundgren and Maybee [Ref. 15], Opsut [Ref. 16], and Roberts [Ref. 17]. Dutton and Brigham [Ref. 14:pp. 315-317] were first to introduce the case in which D was not necessarily an acyclic digraph. They also went on to characterize competition graphs of digraphs that were allowed to have loops. Roberts and Steif [Ref. 18:pp. 323-325] studied parallel characterizations for the case in which the digraphs were allowed to have cycles but no loops.

1. Assumptions

The case in which the digraphs were allowed to have cycles but no loops is the case studied in this chapter. This is due to the fact that communication networks modeled by digraphs most commonly fall into this class of digraphs. A loop would imply that a station was linked or communicating to itself, which does not make sense. Furthermore, to allow for an exchange of information, the network should have cycles. This gives stations in the network alternative routes over which to communicate to other stations. These alternatives help make the overall network more reliable and/or efficient. Therefore, the first two assumptions made about the digraph model of the communication network under study, is that it should be loopless but could have cycles.

The next assumption made involves competition numbers. Digraphs with competition graphs having a competition number greater than zero will not be considered. In other words, given a conflict graph on n vertices, the assumption implies that the digraph must also have n vertices. (Formally, the term competition number is not appropriate for a competition graph of a digraph with cycles, as used here. However, since these types of competition graphs are not studied, the formalities about a competition number are not discussed further.)

This assumption is a logical one considering that a decision-maker would know all of the stations in the network. Thus, the vertex set of the digraph would be known beforehand and would be the same as the vertex set of the corresponding competition graph. Because the competition graph under discussion is actually a conflict graph, from this point on we will use the term conflict graph along with the notation C(D).

The last assumption made about the model is that the conflict graph is connected. This assumes that the conflict graph for the original communications network has been separated into different components, and each is a separate problem in itself. The algorithm will only apply to each separate component of the conflict graph. Thus, when the term conflict graph, digraph, and network are used in describing the model, they are actually referring to the one specific component currently under study. The assumptions made concerning the model are summarized in Table 3.2.

TABLE 3.2

ASSUMPTIONS

#1 - The digraph modeling the network does not have loops.

#2 - The digraph may have cycles.

#3 - Competition graphs (conflict graphs) having a positive competition number are not considered.

#4 - The conflict graph is connected.

2. Approaching the Problem

The idea to study a communication network from the worst case approach arose while studying the conflict graph of a network. The real question is: Given the conflict graph of a digraph, how complex can the digraph be, with respect to the number of arcs it contains? In other words, how complex can the actual network be and still possess the exact characteristics of the original conflict graph? Thus, given a conflict graph, the corresponding digraph will be constructed so that it contains the maximum number of arcs. This "maximum" digraph will still possess only the conflicts implied by the conflict graph.

Once the maximum digraph is found, one can see how complex the actual network might be. In this manner, a decision-maker is able to identify how many additional links can be supported by the network, without increasing the current number of conflicts. Thus, the overall network can be made more reliable or efficient without requiring additional frequency support. So the number of frequencies assigned, based upon the conflict graph, are not affected, yet more links can be added to the original network.

The concept of a conflict graph is a flexible tool which proves to be very valuable to model problems associated with communication networks. The flexibility results from the fact that the "conflict" which is modeled by a conflict graph, can model different parameters or constraints in the network. For instance, the conflict can stand for an adjacent channel restriction between two stations in one case, or it can model cochannel constraints in another application. Thus, the conflict graph is a tool which can be used to model more than one specific constraint, although not in the same model.

E. ALGORITHM

1. Overview

To maximize a digraph, D, we begin in a trial and error type fashion with conflict graphs on three vertices. All of the different graphs with three vertices are investigated before studying graphs with four or more vertices. The goal is to put as many arcs as possible into the corresponding digraph without violating the conditions implied by the conflict graph. Because the method used maximizes the number of arcs drawn into each vertex, the focus to maximize D is changed from the digraph as a whole, to the vertex level. Obviously, maximizing the number of arcs in the overall digraph is equivalent to maximizing the number of arcs going into each vertex, since you cannot send more than one maximal clique into a vertex. Before going into any more detail about the analysis which leads to the algorithm employed, there are certain mathematical preliminaries that should be covered.

2. Mathematical Preliminaries

This section describes two theorems that are used in support of the algorithm derived. The theorems discussed here appear exactly as they were originally presented. Thus, when the word competition graph is used, there is nothing lost in the translation if one were to use conflict graph instead.

This is due to the fact that a conflict graph is nothing more than a specific case of the competition graph.

The first theorem, attributed to Roberts and Steif [Ref. 18:p. 324], explains certain characteristics which a competition graph (conflict graph) must possess in order for it to be the competition graph of a digraph without loops. Roberts and Steif use the notation m(G) for the clique covering number, whereas the notation cc(G) is more widely used currently.

Theorem 1 (Roberts and Steif). If $|V(G)| = n$, then G is a competition graph of a digraph which has no loops if and only if $G \ncong K_2$ and $m(G) \leq n$.

Since the digraph, D, is not allowed to have loops, G cannot be K_2 because it is not possible to construct D without loops. D would have to look like Figure 3.5 below in order to have K_2 as its competition graph.

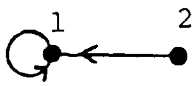
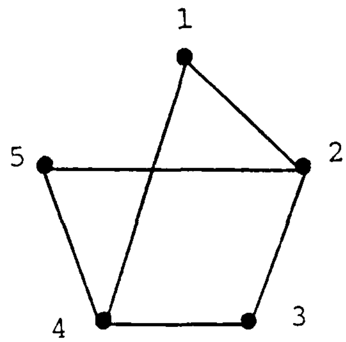


Figure 3.5 Digraph

If G has more than n cliques in its edge clique covering (ECC), then it will not be possible to construct D on n cliques. For example, consider the competition graph in Figure 3.6. The ECC is also given in the figure. Here, the

ECC consists of six K_2's. Since n is 5 in this graph, it is not possible to draw a digraph on five vertices which satisfies the six separate conflicts implied by the six cliques of G.



ECC: $C = \{C_1, C_2, \ldots, C_6\}$

$C_1 = \{1,2\}$

$C_2 = \{2,3\}$

$C_3 = \{3,4\}$

$C_4 = \{4,5\}$

$C_5 = \{2,5\}$

$C_6 = \{1,4\}$

Figure 3.6 A Competition Graph and its ECC

The best one could do is satisfy five edges of G corresponding to five of the six cliques in its ECC (see Figure 3.7 below). Figure 3.7 displays other information in a way that will be used throughout the rest of this chapter. The clique {4,5} shown associated with vertex 1 of the digraph, implies that vertex 1 has been used in satisfying the conflict corresponding to that clique in the conflict graph. So, clique {4,5} was "sent" to vertex 1 in this case. By sending a clique into a vertex, u, it is meant to draw an arc (t,u) in D, for all vertices t that are elements of the clique. Therefore, Figure 3.7 is not capable of satisfying the sixth clique, {2,5}. This also gives an example of a

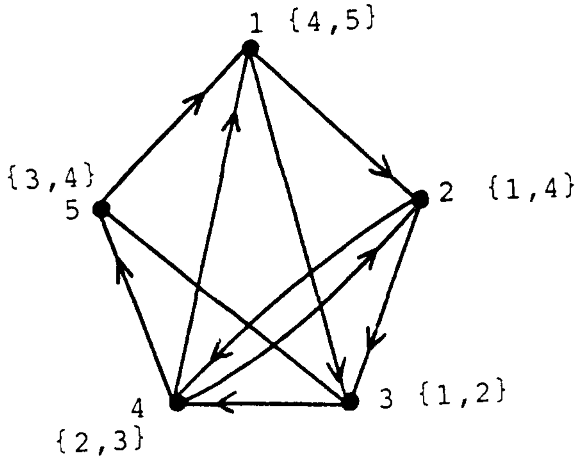


Figure 3.7 Digraph

graph which requires additional isolated vertices to become a competition graph, i.e., its competition number is not equal to zero.

This has been a short explanation of what Theorem 1 actually implies. Roberts and Steif [Ref. 18:pp. 324-325] have presented a formal proof of the theorem, which is based on the following theorem.

Theorem 2 (Roberts and Steif). G is a competition graph of a digraph which has no loops if and only if there are cliques $C_1, C_2, \ldots, C_p$ which cover the edges of G and such that if $D_i = V(G) - C_i$, then $\{D_1, D_2, \ldots, D_p\}$ has a system of distinct representatives.

Theorem 2 plays a major part in the algorithm to follow, since it guarantees that the family of sets $\{D_1, D_2, \ldots, D_p\}$ has an SDR, if the D_i are constructed as defined.

The proof of Theorem 2 can be found in Roberts and Steif [Ref. 18:p. 324] and is a very elegant, but simple proof.

3. Formulation and Analysis of the Algorithm

As the analysis continued, it was soon realized that the maximum number of arcs that could be sent into a vertex was associated with the maximal cliques of C(D). For every pair of arcs (x,v) and (y,v) that enter vertex v of D, there must exist an edge {x,y} in C(D). If there were three or more arcs into vertex u in D, then there has to exist an edge in C(D) between every pair of vertices which enter u. For example, consider the conflict graph and corresponding digraph in Figure 3.8. In order to add arcs (x,v), (y,v), and (z,v) to D, there must exist edges {x,y}, {y,z} and {x,z} in C(D), which is a 3-clique. Therefore, it is not possible to add arcs from vertices $v_1,v_2,\ldots,v_k$ to vertex z in D unless C(D) contains a k-clique made up of vertices $v_1,v_2,\ldots,v_k$.

Figure 3.8 A Conflict Graph and its Digraph

The idea of using maximal cliques naturally goes along with the goal of trying to maximize the number of arcs in D. In order to maximize the number of arcs into each vertex, the maximal cliques of C(D) have to be known beforehand. Thus, the first requirement is to find all of the maximal cliques in C(D). Therefore, an edge clique covering is obtained for C(D) using only maximal cliques. This part of the problem is a major task in itself for arbitrary graphs. In fact, the problem of covering the edges of a graph by the minimal number of cliques is known to be NP-complete.

By knowing the maximal cliques, it is easy to determine the maximum number of arcs that could enter a single vertex in D. Obviously, since the maximum is sought, the largest maximal clique is the place to start.

An algorithm is derived to organize the steps to be performed in arriving at this maximum digraph. But before an algorithm can be developed, the problem has to be bounded first. The lower bound on the maximum number of arcs is derived and then the upper bound is found.

The algorithm proceeds as follows. Given the conflict graph, C(D), we find the minimal ECC using maximal cliques, C_i, and construct $D_i = V(G)-C_i$. Let n be the number of vertices in C(D), m be the number of cliques in the ECC, and C_m be the clique of largest cardinality. Define $k(G)$ to be the size of this largest clique. (The restriction that $m \leq n$ still applies.) By Theorem 2, we know an SDR exists for $F = \{D_1, D_2, \ldots, D_m\}$, since C(D) is a competition graph. Consider the SDR for $D_1, D_2, \ldots, D_m$. Construct arcs in D by sending cliques

C_i into the vertex selected as the representative for D_i. At this point, the digraph contains $\sum_{i=1}^{m} |C_i|$ arcs.

The remaining (n-m) vertices in D receive arcs from the vertices in C_m. At worst, each of these vertices, u_i, are elements of C_m. Since D is not allowed to have loops, this implies that $(k(G)-1)$ arcs will be sent into each u_i, where u_i is not a representative for $D_1, D_2, \ldots, D_m$. This adds an additional $(k(G)-1)(n-m)$ arcs to D. Thus, the lower bound for the maximum number of arcs in the digraph corresponding to C(D) is

$$\sum_{i=1}^{m} |C_i| + (k(G)-1)(n-m) \quad \text{for } m \geq 2 \text{ .} \qquad (3.1)$$

Figure 3.9 is an example of a conflict graph which has this lower bound for the maximum number of arcs in its corresponding digraph. The maximum number of arcs is ten, which agrees with Formula 3.1 when n = 4, m = 2 and $k(G)$ = 2.

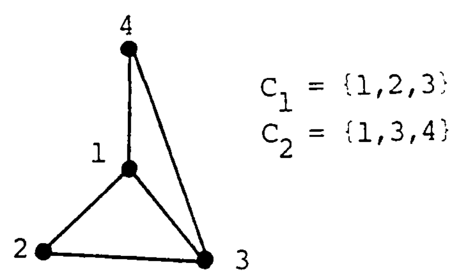


Figure 3.9 Graph

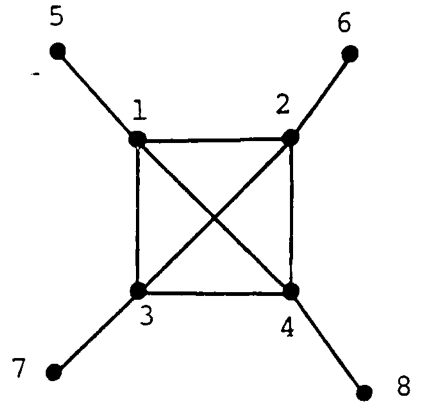
The upper bound on the maximum number of arcs in D is derived in a similar manner. The first part of the bound remains the same, so m vertices in D have a total of $\sum_{i=1}^{m} |C_i|$ arcs. From this point on, the upper bound is different than the lower bound for the remaining (n-m) vertices in D.

The upper bound is achieved when the remaining vertices in D are considered. If these vertices, u_i, are not elements of C_m, then C_m can be sent into each of these vertices without losing any arcs. Thus, an additional $k(G)(n-m)$ arcs are added to D, and the upper bound is

$$\sum_{i=1}^{m} |C_i| + k(G)(n-m) \text{ for } m \geq 2 . \qquad (3.2)$$

Figure 3.10 is an example of a graph in which the maximum number of arcs equals the upper bound. The maximum number of arcs for the corresponding digraph is 24 which is what Formula 3.2 gives for the upper bound, when n = 8, m = 5 and $k(G) = 4$. The graph is able to achieve the upper bound because cliques C_1, C_2, C_3 and C_4 can be sent into vertices 2,1,4 and 3 respectively, which allows C_5 to be sent into the remaining vertices.

One key in reaching the upper bound is that cliques C_1, C_2, C_3 and C_4 are only sent into vertices that are elements of C_5. If a smaller clique is sent into a vertex which is not an element of the largest clique, the upper bound cannot be



$C_1 = \{1,5\}$
$C_2 = \{2,6\}$
$C_3 = \{3,7\}$
$C_4 = \{4,8\}$
$C_5 = \{1,2,3,4\}$

Figure 3.10 Graph

achieved. This observation leads to a conclusion that the upper bound cannot be obtained if $r*k(G) < m-r$, where r is the number of largest maximal cliques with cardinality $k(G)$, and m is the number of maximal cliques. For example, if there are two largest maximal cliques with $k(G) = 3$, and a total of nine cliques, it is not possible to send the seven smaller cliques into seven distinct vertices which are in the two largest cliques. Thus, at least one of the seven smaller cliques must be sent into a vertex outside of the set of the largest cliques. These vertices cannot receive arcs from the largest cliques, so the maximum number of arcs is less than the upper bound. This is just one case in which the maximum digraph would not achieve the upper bound. There are many other cases.

The difference between the upper and lower bounds is (n-m). One should be aware that the bounds are only defined for graphs in which $m \geq 2$. Therefore, a complete graph, K_n, which possesses one maximal clique, is not defined by these bounds.

The algorithm developed to maximize the number of arcs in a digraph given its conflict graph is listed below. An explanation of each step follows the algorithm.

Algorithm (Maximizes the Number of Arcs in a Digraph, Given its Conflict Graph)

Step 1 (Label C(D) and D, Initialize list L)
Label the vertices in C(D) 1,2,...,n.
Label the vertices in D, 1,2,...,n corresponding to the n vertices in C(D).
Put the numbers 1,2,...,n, corresponding to vertices 1,2,...,n in a list L.

Step 2 (Find an ECC for C(D) using maximal cliques)
Find a minimal edge clique covering for C(D) using maximal cliques.
Call the collection of cliques $\mathbf{C} = \{C_1, C_2, \ldots, C_m\}$.

Step 3 (Special case: $C(D) = K_n$)
If C(D) is a complete graph, K_n, then D will be a directed complete graph, DK_n, with a maximum of n(n-1) arcs, stop.
If C(D) is not a complete graph, go to Step 4.

Step 4 (Construct sets D_i and **F**)
Construct the set D_i, for each C_i, found in Step 2, such that $D_i = V(C(D)) - C_i$, $i = 1,2,\ldots,m$.
Let $\mathbf{F} = \{D_1, D_2, \ldots, D_m\}$.

Step 5 (Find an SDR for **F**)
Find a system of distinct representatives, $x_1, x_2, \ldots, x_m$, for **F** out of all possible SDR's which <u>maximizes</u> the number of x_i meeting the following constraint:
Pick $x_i \in D_r$ such that there does not exist D_t for which $x_i \in D_t$ and $|D_r| > |D_t|$.

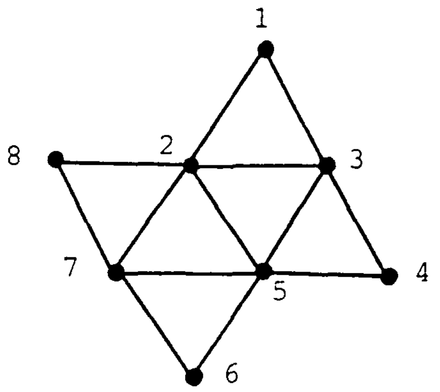
Step 6 (Add arcs to D, Update L)
Create arc (w,x_i) in D for all $w \in C_i$, $i = 1,2,...,m$.
Delete all $x_i \in$ SDR from L. (Now list L has $|n-m|$ vertices remaining in it.)

Step 7 (Maximize arcs into remaining vertices, Update L)
Consider the remaining numbers (vertices) in L in any order.
Pick vertex j from L.
Delete j from all cliques C_i in which j is a member and call these new sets R_i. If j is not a member of C_i, the set is still renamed R_i. Do this for $i = 1,2,...,m$.
Pick the largest R_i, with respect to cardinality, breaking ties randomly, and call this new set K_j.
Create arc (z,j) for all $z \in K_j$.
Delete vertex j from list L.

Step 8 (Repeat Step 7)
If L is not empty, go to Step 7.
If L is empty, stop. All vertices in D have been assigned a maximum number of incoming arcs, and D is maximum.

The first step in the algorithm is the initialization step. Since one assumption of the model is that the conflict graph must have competition number zero, then $|V(C(D))|$ must equal $|V(D)|$. In other words, if C(D) has n vertices, then so does the corresponding digraph. Therefore, Step 1 labels the vertices of both C(D) and D with the labels $1,2,...,n$ and puts the vertices in a list.

Step 2 of the algorithm finds a minimal ECC for C(D) using maximal cliques. It should be noted here that this minimal ECC is not necessarily unique for C(D). Figure 3.11 is an example of a graph which has two different minimal ECC's, using maximal cliques. The two different coverings, C



C = {C1,C2,C3,C4}   C' = {C_1,C_2,C_3,C_4}

C_1 = {1,2,3} C_1' = {1,2,3}

C_2 = {3,4,5} C_2' = {3,4,5}

C_3 = {5,6,7} C_3' = {5,6,7}

C_4 = {2,7,8] C_4' = {2,7,8}

C_5 = {2,5,7} C_5' = {2,3,5}

Figure 3.11 A Graph with Two Different Minimal Coverings

and C', are listed in the figure. The difference in the coverings is due to the fact that edge {2,5} can be considered an edge in clique {2,5,7} or {2,3,5}. This choice for an ECC will not affect the maximization of digraph D, which is accomplished by the remaining steps in the algorithm.

In arriving at this algorithm, several different conflict graphs were studied, which helped to develop certain steps in the algorithm. For instance, Figure 3.6 was the first conflict graph studied in which there were more maximal cliques than vertices. By Theorem 1, the conflict graph is not the competition graph of a digraph without loops, so the assumptions of the model are not met. As a result, Step 2 requires that the number of maximal cliques in the minimal ECC be less than or equal to the number of vertices in the conflict graph. Assumption #3 is also derived as a result of this analysis.

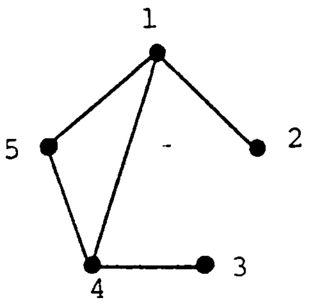
The third step handles the special case when the conflict graph is a complete graph, K_n. If C(D) is complete, then digraph D is a directed complete graph, DK_n, with n(n-1) arcs, because this is the maximum number of arcs possible in a digraph without loops, and it satisfies the conflict graph. If C(D) is not complete, the algorithm proceeds.

Once the minimal ECC is found, it is time to decide how the maximal cliques of the ECC are to be used to maximize the digraph. Because D is not allowed to have loops, it is not possible to send a maximal clique into a vertex that is an element of that maximal clique. (Once again, sending a clique, C_x, into a vertex, y, means drawing an arc (v,y) in digraph D, for every vertex v ϵ C_x, which is a clique of C(D)). Consequently, all of the conflicts corresponding to this clique are not satisfied in this case. In order to maintain maximization from a maximal clique, C_r, that clique has to be sent into a vertex which is not an element of C_r. Step 4 helps satisfy this requirement by constructing set D_i for each C_i ϵ C, such that D_i is defined to be $V(C(D))-C_i$. This implies that clique C_i can be sent into any vertex which is an element of D_i without losing an arc in digraph D.

The next step helps to decide which maximal clique can be sent into which vertex. Another factor becomes apparent at this point, in addition to maintaining the maximization from the cliques. In order to satisfy the conflicts of C(D), each maximal clique has to be sent into a <u>distinct</u> vertex. If more than one maximal clique can be sent into the same vertex, then at least two of the maximal cliques in the ECC are not maximal to begin with. Therefore, a specific vertex cannot receive more than one maximal clique. Furthermore, in maximizing D, it makes sense to send the largest maximal clique into as many vertices as possible.

Since each maximal clique requires a distinct vertex, the idea of using a system of distinct representatives (SDR) is analyzed. The SDR satisfies this requirement specifically. However, the SDR does not satisfy the requirement of sending the largest maximal clique into as many vertices as possible, without modifying it in some way. (An SDR is guaranteed by Theorem 2, since the D_i are defined as specified.)

When the conflict graph in Figure 3.12 is studied, the SDR is modified to ensure that digraph D remains maximized. If the SDR 3,4,2 is chosen for **F**, an arc would be lost in digraph D. This happens because of sending clique C_3 into vertex 2 instead of sending clique C_1 into vertex 2. C_3 has only two elements in its set, where C_1 has three elements. An optimal SDR would be 3,4,1. As a result, the SDR is modified to ensure that each vertex receives the largest maximal clique possible. Consequently, the following additional restriction is added to the SDR. Select an SDR, $x_1, x_2, \ldots, x_m$ which <u>maximizes</u> the number of x_i where $x_i \in D_r$ such that there does not exist D_t for which $x_i \in D_t$ and $|D_r| > |D_t|$. Figure 3.12 is



$C = \{C_1, C_2, C_3\}$

$C_1 = \{1,4,5\}$

$C_2 = \{1,2\}$

$C_3 = \{3,4\}$

$F = \{D_1, D_2, D_3\}$

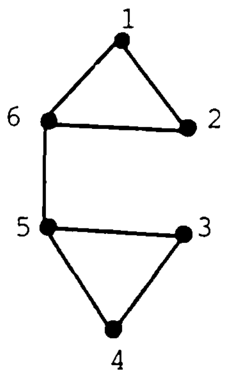
$D_1 = \{2,3\}$

$D_2 = \{3,4,5\}$

$D_3 = \{1,2,5\}$

Figure 3.12 Example Conflict Graph

an example in which all three of the x_i meet this restriction. However, there are examples of conflict graphs in which all m of the x_i cannot meet this restriction. Figure 3.13 is one example. In this example, it is not possible to select an SDR in which all three of the x_i meet the restriction. This is due to the fact that x_3 must be chosen from among vertices 1,2,3 or 4 which are all elements of a smaller D_i.



$C = \{C_1, C_2, C_3\}$

$C_1 = \{1,2,6\}$

$C_2 = \{3,4,5\}$

$C_3 = \{5,6\}$

$F = \{D_1, D_2, D_3\}$

$D_1 = \{3,4,5\}$

$D_2 = \{1,2,6\}$

$D_3 = \{1,2,3,4\}$

Figure 3.13 Example Conflict Graph

Once this step is completed in the algorithm, the conflicts in C(D) are all met. This is referred to as meeting the "minimum requirements" of the graph. If the additional restriction is not met in selecting an SDR for **F**, the minimum requirements are still satisfied, but the digraph will not have as many arcs. After the selection of an optimal SDR, it is time to look at the remaining (n-m) vertices in digraph D.

Steps 7 and 8 are used to maximize the remaining vertices which are not chosen by the SDR in Step 5. The goal is to send the largest maximal clique into each of these vertices, if possible. Consider vertex j which was not picked as a representative of the SDR. Since loops are not allowed in the digraph, this is taken into account by deleting element j from all of the maximal cliques in which j is a member. If j is not a member of a maximal clique, the clique remains the same. Now, the largest of these sets is chosen, so that the elements of the set will be sent into vertex j in the digraph. This step ensures that the largest maximal clique is sent into each of the vertices not selected in the SDR.

This completes the analysis of the algorithm. The next section applies the algorithm to an example network in order to show how the steps are completed in detail.

F. APPLYING THE ALGORITHM TO A COMMUNICATIONS NETWORK

In this section, we apply the algorithm described in the previous section to an example communications network and arrive at a maximum digraph. This maximum digraph, D, is a digraph that satisfies the given conflict graph, C(D), and which possesses the maximum number of arcs possible. Therefore, we start with the conflict graph that corresponds to the digraph that models the network. Let us consider the conflict graph and its corresponding digraph shown in Figure 3.14 below as the example. Starting with Step 1, the algorithm is applied to C(D). Step 1 labels both C(D) and D and puts the vertices 1,2,...,7 into a list L. Digraph D will

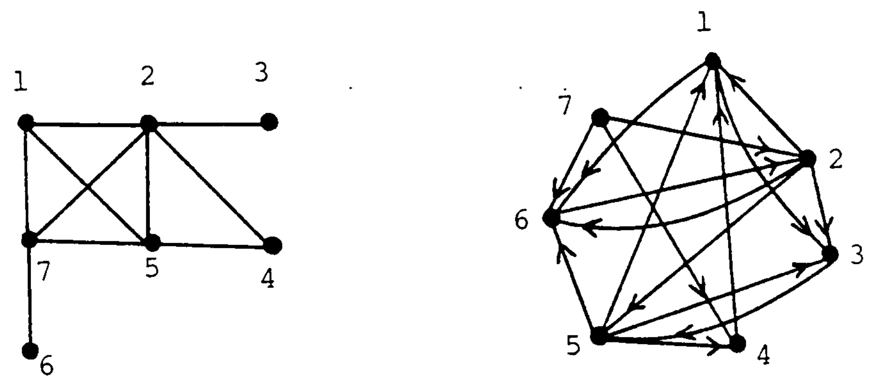
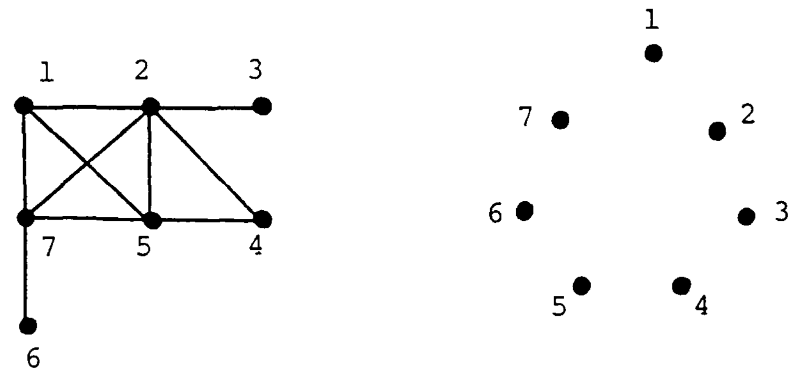


Figure 3.14 Conflict Graph and Digraph of the Example Network

now refer to the maximum digraph and not the original digraph above. The results shown in Figure 3.15 are obtained from Step 1. Once Step 1 is completed, a minimal ECC is found using maximal cliques.

The minimal ECC for C(D) is in fact unique for this example and consists of four maximal cliques. Hence, C = $\{C_1, C_2, C_3, C_4\}$, where $C_1 = \{1,2,5,7\}$, $C_2 = \{2,4,5\}$, $C_3 = \{2,3\}$ and



$L = \{1,2,3,4,5,6,7\}$

Figure 3.15 Results of Step 1

$C_4 = \{6,7\}$. Note here that $m = 4$ and $n = 7$, so $m \leq n$ is satisfied.

Since C(D) is not a complete graph, K_7, the algorithm proceeds to Step 4. At Step 4, the sets D_1, D_2, D_3 and D_4 are constructed for C_1, C_2, C_3 and C_4 respectively. The family of sets, **F**, is then constructed with D_1, D_2, D_3 and D_4 as its elements. When Step 4 is completed, the results shown in Table 3.3 are available. The next step can be considered the most important part of the entire algorithm. This step involves finding an SDR for **F**.

TABLE 3.3

RESULTS FROM STEP 4

$\mathbf{C} = \{C_1, C_2, C_3, C_4\}$	$\mathbf{F} = \{D_1, D_2, D_3, D_4\}$
$C_1 = \{1,2,5,7\}$	$D_1 = \{3,4,6\}$
$C_2 = \{2,4,5\}$	$D_2 = \{1,3,6,7\}$
$C_3 = \{2,3\}$	$D_3 = \{1,4,5,6,7\}$
$C_4 = \{6,7\}$	$D_4 = \{1,2,3,4,5\}$

Step 5 of the algorithm selects an SDR for **F** which must meet the additional restriction mentioned in the previous section. As a result of meeting the additional restriction, 3, 4 or 6 could be a representative for D_1, 1 or 7 could be a representative for D_2, 5 is the only choice as a representative for D_3, and 2 or 5 are choices for D_4. Since the SDR requires distinct representatives, 5 is not then a choice for D_4 because it must be chosen for D_3. Therefore, there are six possible SDR's for **F** in which all of the representatives meet the restrictions. They are given in Table 3.4 below. Each of

TABLE 3.4

POSSIBLE SDR'S FOR F

SDR
#1 - 3,1,5,2
#2 - 4,1,5,2
#3 - 6,1,5,2
#4 - 3,7,5,2
#5 - 4,7,5,2
#6 - 6,7,5,2

these SDR's will result in a maximum digraph with the same number of arcs. Since there is a choice in the SDR for this case, D is not unique for this example. Without loss of generality, SDR #1 will be used to complete this example. Once the SDR is selected, the corresponding arcs must be drawn in D and list L is updated. The list now contains elements 4, 6 and 7 and D looks like the digraph shown in Figure 3.16, after Step 6 is completed. Once again, a clique drawn next to a vertex implies the elements (vertices) in that clique have arcs into that vertex.

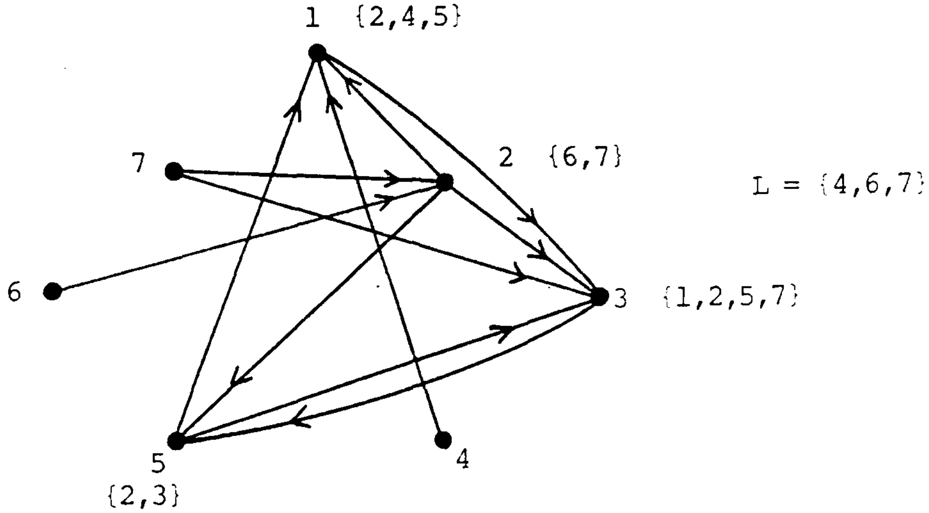


Figure 3.16 Digraph Through Step 6

At this point, the digraph in Figure 3.16 satisfies all the minimum requirements of the conflict graph. All that remains is to maximize vertices 4, 6 and 7 using Step 7. Consider first vertex 4. In Step 7, vertex 4 is deleted from the C_i for which it is an element, and the sets are renamed R_i. Those cliques that do not contain vertex 4 as an element are also renamed. As a result of Step 7, the sets R_i are those found in Table 3.5. Set R_1 is the largest in this example, so arcs are drawn in D from each vertex that is an element of R_1 to vertex 4, and vertex 4 is deleted from list L. Since L

TABLE 3.5

SETS R_i

$R_1 = \{1,2,5,7\}$
$R_2 = \{2,5\}$
$R_3 = \{2,3\}$
$R_4 = \{6,7\}$

still has two elements in it, Step 7 must be repeated again for vertices 6 and 7. When vertex 6 is considered, the sets R_i are exactly the same as when vertex 4 was considered. Thus, the arcs (1,6), (2,6), (5,6) and (7,6) are added to D at this point. The last vertex to be considered is vertex 7. In constructing sets R_i, there are two sets, R_1 and R_2, with three elements in them, so another choice is introduced. One could choose $R_1 = \{1,2,5\}$ or $R_2 = \{2,4,5\}$ as the largest set. In this example, R_1 is chosen and arcs (1,7), (2,7) and (5,7) are added to D.

Finally, the algorithm is completed and D is maximized. The maximum number of arcs that digraph D may contain is 22, and looks like that of Figure 3.17, based on the choices made in this example.

G. SUMMARY AND CONCLUSIONS

In conclusion, a maximum digraph is found for the given conflict graph of Figure 3.14, using the algorithm of the previous section. This digraph is not unique since there are choices made at Steps 5 and 7 of the algorithm. There are six

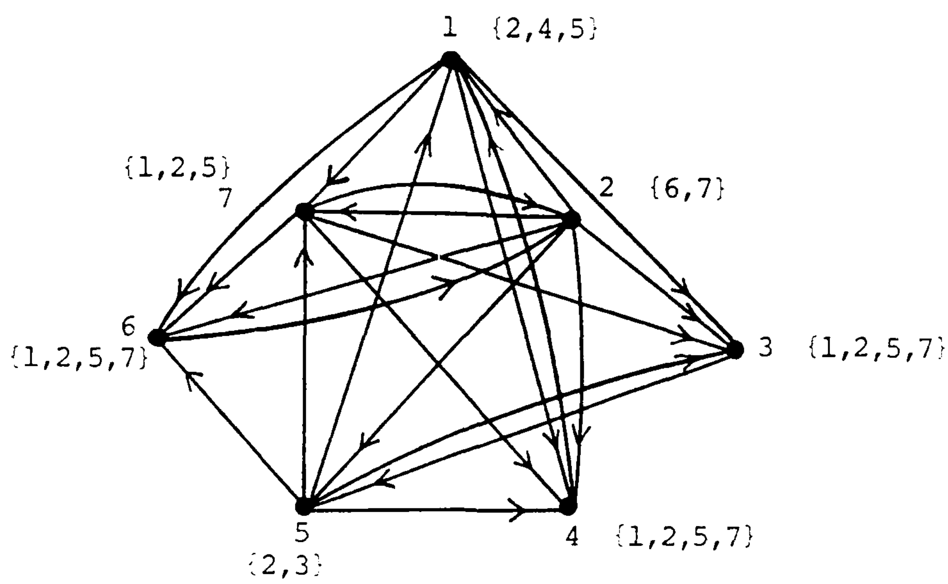


Figure 3.17 Maximum Digraph

possible SDR's at Step 5 and two choices for vertex 7 at Step 7. It is also possible for a choice to arise in selecting an ECC in Step 2, although this example had a unique covering.

Although Figure 3.17 is a maximum digraph for the given conflict graph, it does not achieve the upper bound. The upper bound for this graph would be 23 arcs since $n = 7$, $m = 4$, and $k(G) = 4$. The difference arises from the fact that C_1 is sent into a vertex that is an element of C_1.

This maximum digraph provides certain information to the decision-maker. If the current communication network being analyzed possesses 22 links, then no additional links can be added without affecting the given conflict graph. If the conflict graph is changed, the frequency assignment for the network may or may not change, since the frequency assignment is found using the conflict graph. If the communication network actually has fewer than 22 links in it, then it is possible to add links to the network in such a way that the current frequency assignment is not affected.

In order to determine the specific links that could be added, the decision-maker must examine both the conflict graph and corresponding digraph that models the network. In this way, the choices made when applying the algorithm would be picked to coincide with the existing network. Thus, the additional links would be determined from information obtained from this maximum digraph corresponding to the actual network.

IV. CONCLUSIONS AND REMARKS

As the radio frequency communication environment grows more complex each day, it is important to investigate new ways to increase the efficiency of communication systems. This thesis looks at one subproblem of managing a communication network which requires finding a frequency assignment for the network. Provided that an acceptable frequency assignment can be found using techniques outlined earlier, this thesis is concerned with examining a problem which succeeds the frequency assignment phase. That problem involves placing additional links in the network so as to increase network efficiency and/or reliability, while not affecting the current number of frequencies assigned to the network.

Because of certain characteristics associated with graphs and digraphs in conjunction with networks, this thesis approaches the problem using graph-theoretic principles. The overall concept is to model a communication network with a digraph and then look at the conflict graph of this digraph. By coloring this conflict graph, one can obtain a frequency assignment for the network. But once a network possesses a frequency assignment, is it still possible to improve the efficiency of the network without affecting the frequency assignment? That is the problem examined in this thesis. By placing additional links which do not alter the number of frequencies assigned into a network that does not have a maximal number of links, the overall network can be made more efficient.

These additional links can be found by comparing the digraph which models the network with the digraph possessing the maximum number of arcs obtained from the conflict graph of the network. The algorithm developed in this thesis helps to achieve this maximum digraph.

The algorithm consists of several steps which contain complex problems in themselves. One involves finding a minimal edge clique covering (ECC) for a conflict graph using only maximal cliques. The problem of finding a minimal ECC in general is an NP-complete problem. This particular problem is currently under investigation. There are other problems which remain unanswered.

One remaining problem is to classify those conflict graphs that correspond to maximum digraphs which achieve the upper bound for the number of arcs it may possess. Another problem involves generalizing the algorithm so that it finds the maximum digraph for any conflict graph. This problem would most likely center around Step 5 of the algorithm which selects a system of distinct representatives for the D. Another possible problem would be to classify those graphs which possess a unique minimal ECC using maximal cliques only. It will also be necessary to examine decomposing large networks into smaller subnetworks so that algorithms can be used to maximize less complex systems. This is due to the problem being NP-complete. If it is to be solved in an acceptable amount of time, it is possible that a series of reduced subnetworks will make that task easier.

In addition to these specific problems, there are other general communication problems that could still be examined. One of these problems would be to find a site for a future station in the network whose mission is solely to increase the efficiency of the network. This type of station is closely associated with what is called a retransmission station which is currently being used.

These are just a few of the many problems related to communication networks and systems which are still under study. Because of the overall complexity of the communication environment and the many factors involved, there will always be new ways in which the utilization of our electromagnetic spectrum can be made more efficient.

LIST OF REFERENCES

1. Roberts, F.S., Discrete Mathematical Models, Prentice-Hall, Inc., 1976.

2. Aho, A.V., Hopcroft, J.E., and Ullman, J.D., Data Structures and Algorithms, Addison-Wesley Publishing Co., 1983.

3. Garey, M.R., and Johnson, D.S., Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman and Co., 1979.

4. Mathur, K., and others, "Applications of Integer Programming in Radio Frequency Management," Management Science, V. 31, July 1985.

5. Allen, J.D., Helgason, R.V., and Kennington, J.L., "The Frequency Assignment Problem: A Solution via Nonlinear Programming," Naval Research Logistics, V. 34, February 1987.

6. Hale, W.K., "Frequency Assignment: Theory and Applications," Proceedings of the IEEE, V. 68, December 1980.

7. Zoellner, J.A., and Beall, C.L., "A Breakthrough in Spectrum Conserving Frequency Assignment Technology," IEEE Transactions on Electromagnetic Compatibility, V. 19, August 1977.

8. Bazaraa, M.S., and Jarvis, J.J., Linear Programming and Network Flows, John Wiley & Sons, Inc., 1977.

9. Cameron, S.H., "The Solution of the Graph-Coloring Problem as a Set-Covering Problem," IEEE Transactions on Electromagnetic Compatibility, V. 19, August 1977.

10. Roberts, F.S., Applied Combinatorics, Prentice-Hall, Inc., 1984.

11. Dossey, J.A., and others, Discrete Mathematics, Scott, Foresman and Co., 1987.

12. Baybars, I., "Optimal Assignment of Broadcasting Frequencies," European Journal of Operational Research, V. 9, 3 March 1982.

13. Cohen, J.E., "Interval Graphs and Food Webs: A Finding and a Problem," Rand Corporation Document 17696-PR, Santa Monica, California, 1968.

14. Dutton, R.D., and Brigham, R.C., "A Characterization of Competition Graphs," Discrete Applied Mathematics, V. 6, 1983.

15. Lundgren, J.R., and Maybee, J.S., "A Characterization of Competition Graphs of Competition Number m," Discrete Applied Mathematics, V. 6, 1983.

16. Opsut, R.J., "On the Computation of the Competition Number of a Graph," SIAM Journal on Algebraic and Discrete Methods, V. 3, December 1982.

17. Roberts, F.S., "Food Webs, Competition Graphs, and the Boxicity of Ecological Phase Space," Theory and Applications of Graphs, Springer-Verlag, 1978.

18. Roberts, F.S., and Steif, J.E., "A Characterization of Competition Graphs of Arbitrary Digraphs," Discrete Applied Mathematics, V. 6, July 1983.

BIBLIOGRAPHY

Corneil, D.G., and Fonlupt, J., "The Complexity of Generalized Clique Covering," Discrete Applied Mathematics, V. 22, 1989.

Garey, M.R., and Johnson, D.S., "The Complexity of Near-Optimal Graph Coloring," Journal of the Association for Computing Machinery, V. 23, January 1976.

Golumbic, M.C., Algorithmic Graph Theory and Perfect Graphs, Academic Press, Inc., 1980.

Headquarters, Department of the Army, Army Regulation No. 5-12, Army Management of the Electromagnetic Spectrum, 15 June 1983.

Karp, R.M., "On the Computational Complexity of Combinatorial Problems," Networks, V. 5, January 1975.

Lawler, E.L., "Algorithms, Graphs, and Complexity," Networks, V. 5, January 1975.

Middlekamp, L.C., "UHF Taboos--History and Development," IEEE Transactions on Consumer Electronics, V. 24, November 1978.

Misra, J., "Necessary Subproblems in Maximal Clique Construction," Proceedings of the 7th Hawaii International Conference on System Sciences, Western Periodicals Co., 1974.

Orlin, J., "Contentment in Graph Theory: Covering Graphs with Cliques," Proceedings of the Koninklijke Nederlandse Academie van Wetenschappen, Amsterdam, V. 80, 1977.

Pullman, N.J., Shank, H., and Wallis, W.D., "Clique Coverings of Graphs V. Maximal-Clique Partitions," Bulletin of the Australian Mathematical Society, V. 25, 1982.

Pullman, N.J., "Clique Covering of Graphs IV. Algorithms," SIAM Journal on Computing, V. 13, February 1984.

Raychaudhuri, A., and Roberts, F.S., "Generalized Competition Graphs and Their Applications," Methods of Operations Research, Anton Hain, Konigstein, West Germany, 1985.

Schrage, L., Linear Programming with LINDO, Scientific Press, Palo Alto, California, 1980.

Schrage, L., User's Manual for LINDO, Scientific Press, Palo Alto, California, 1980.

Scott, D.D., "The Competition-Common Enemy Graph of a Digraph," Discrete Applied Mathematics, V. 17, June 1987.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Commandant of the Marine Corps Code TE 06 Headquarters, U.S. Marine Corps Washington, D.C. 20380-0001	1
4.	Chairman, Code MA Department of Mathematics Naval Postgraduate School Monterey, California 93943-5000	1
5.	Professor Kim Hefner, Code MA Hk Department of Mathematics Naval Postgraduate School Monterey, California 93943-5000	3
6.	Professor John Thornton, Code MA Th Department of Mathematics Naval Postgraduate School Monterey, California 93943-5000	2
7.	Lt Douglas E. Hintze, USN Strategic Systems Programs, SP-1532 Washington, D.C. 20376	1
8.	Lt Randy D. Black, USN U.S. Naval Test Pilot School, Class 97 NAS Patuxent River, Maryland 20670	1
9.	Captain Donald W. Hintze, USMC Box 21 Highway 5 South Waldorf, Maryland 20601	3