GRAPH COLOURING AND SCHEDULING

AN APPROACH TO NURSES SCHEDULING

BY

KWAME GYAMFI OKRAH BSc. (HONS) MATHEMATICS

A THESIS SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL MATHEMATICS

KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN

PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

INSTITUE OT DISTANCE LEARNING

MAY, 2009

DECLERATION

I hereby declare that this submission is my own work towards the MSc and that, to the best of my knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any other degree of the University, except where due acknowledgement has been made in the text.

KWAME GYAMFI OFRAH 03/03/2010

Student Name & ID **Signature** **Date**

Certified by

DR. S.K Amponsah 07/04/10

Supervisor Name **Signature** **Date**

Certified by

DR. S.K Amponsah 07/04/10

Head of Mathematics Department **Signature** **Date**

Certified by

Prof E. Badu 08/04/2010

Dean of Institute of **Signature** **Date**

Distance Learning

ABSTRACT

The ever growing rate of infant and maternal mortality in districts, municipal, teaching, regional and military Hospitals of Ghana have made the optimal assignment of nurses to shifts, a problem of primary importance. We investigate the various soft constraints of each nurse and its implications and conflicts. Graph colouring is applied. The investigation leads to the creation of conflict graph for the nurses. The conflict graph is coloured using the greedy algorithm approach. This results to a conflict free graph for the nurses. We therefore applied the hospital hard constraints to schedule the various nurses at the maternity ward of Ejura District Hospital to the available shifts, namely Morning, Afternoon and Evening shifts.

ACKNOWLEDGMENT

This thesis has been a real odyssey. It wouldn't have come to reality without the creator and sustainer of this universe. He is truly a ubiquitous one. Greatest thanks surely go to God Almighty.

Great thanks go to my supervisor Dr. S.K. Amponsah for his encouragement, support and most of all expertise. Your fatherly care and advice will forever be remembered. In fact you are second to none. May God richly bless you, your family and friends.

Heartfelt gratitude goes to the principal midwifery officer (matron) of the maternity ward of Ejura District Hospital. To all nurses of the maternity ward, we say thanks.

Similarly, appreciation goes to Miss Dorothy Aduako Sarpong of Kwame Nkrumah University of Science and Technology Hospital, Kumasi for her support to us.

We cannot mention the names of all those who supported us through this project but we want to express our whole-hearted appreciation to the Administrator of Sekyedumase Senior High School, Miss Charlotte Arhin, Mr. Victor Owusu Ansah of Ejuraman Senior High School and to all our loved ones for their spiritual and mental support.

We wish to thank all lectures/staff of the Institute of Distance Learning especially Dr. Tamakloe and Ben. It's been great being a part of the Institute fraternity. We are really proud to have been under your tutelage.

And to all whose names have not been mentioned here, we say a big, big thank you. You are too wonderful.

TABLE OF CONTENTS

Chapter One

Chapter Two

Chapter Three

Chapter Four

Chapter Five

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

1.1 BACKGROUND TO THE STUDY

The investigation of graph colouring originally rose from the well-known four-colour theorem, which was posed as a conjecture in the 1850s [1][25].The four- colour theorem asks the question; whether the regions of any map, for instance the map of Kumasi Township could be coloured with four colours so that suburbs with common border have different colours. Many incorrect proofs of the four-colour theorem were published often with hard-to-find errors. However, it was finally proved by the American Mathematicians; Appel and Haken in 1976 [2].

Ever since and even before, graph colouring has been studied extensively and there are several interesting practical and feasible problems that can be modeled by graph colouring. The surge in recent times has resulted in countless real world problem applications, which includes; Time tabling Scheduling problems, Frequency Assignment, Register allocation, Printed circuit Board testing I and II (Colours and Clique), Analysis of Biological and Archaeological Data and pattern Matching.

For instance with aircraft scheduling, assume that we have β aircrafts and we have to assign them to α flights, where the ith flight is during the time interval (a_i, b_i). Clearly if two flights overlap, then we cannot assign the same aircraft to both flights. The vertices of the conflict graph correspond to the flight; two vertices are connected if the corresponding time intervals overlap. The conflict graph is an interval graph, which can be coloured optimally in polynomial time.

With all the applications of graph colouring, scheduling is the most prominent. Scheduling problems can arise in several different forms and areas of our world. The particular form of scheduling problem required is specific to the institution or environment in which it is needed. For example a hospital schedule problem (duty roster) for medical staff and nurses will be different from a senior high school timetable schedule. Again in the health sector the schedule problem for medical staffs (Doctors and nurses) is totally different from the schedule problem for non-medical staffs (Cleaners, Labourers, Security Officers).

The scheduling problem can be defined as a problem of finding the optimal sequence for executing a finite set of operations (tasks or jobs) under a certain set of constraints that must be satisfied. A scheduler usually tries to maximise the utilization of individuals and/or resources and minimise the time required to complete the entire process being scheduled.

For example, a generic job-shop scheduling problem can be formulated as assigning machines to workers or assigning workers to machines. Similarly, a school timetabling problem can also be seen as allocation of courses to timeslots or allocation of timeslots to courses.

1.2 STATEMENT OF PROBLEM

In a hospital, a new schedule for nurses (duty roster) must be generated for each ward every two weeks or monthly. A hospital ward is an organisational unit that has to fulfil some concrete task, and has both rooms and personnel, the nurses, at its disposal. Usually, the wards of a hospital are completely distinct: each has its own rooms and its own personnel. The schedules of a ward in a hospital can be done separately.

We consider the scheduling problem for the maternity ward at Ejura District Hospital. In general a maternity ward consist of about 10 – 20 health workers (staff) having different qualifications and responsibilities. These staff are placed into categories based on their qualifications, experience and job description which includes principal midwifery officers (PMO), senior midwifery officers (SMO), midwifery officers (MO), senior staff midwifes (SSM), staff midwifes (SM), part-time midwifes (PTM), student midwifes (SM) and midwifery aids (MA). Some of the nurses can replace people from another category (depending on their qualifications). Each replacement by a person from another category will raise the evaluation function by an amount.

In Ghana, nurses work in shifts. Generally, there are three basic shifts in a day, namely the Morning or (AM) shift (M), the Afternoon or PM shift (A) and Evening or Night shift (N). An irregular shift (I) has a special working hours either arrange by the nursing officer or the midwifery officer. Nurses are entitled to different types of holidays. These include day-off (D/O), compensation-off (C/O), public holiday-off (P/O) and vacation leave (VL). Nursing roastering is concerned with highly constrained resource allocation problems. The work head needs to be assigned to nurses periodically taking into account a number of constraints and requirement. Basically we consider two types of constraints namely the Hard constraints (imperative planning rules) and the Soft constraints (preferences planning rules).

Hard constraints are those that must be satisfied in order to have a feasible schedule due to physical resource restrictions and legislation. When requirements are desirable but not obligatory they are referred to as Soft constraints. Soft constraints are often used to evaluate the quality of feasible schedules. In generating the roaster, we must ensure that every planning decision made is coherent with the hard rules.

Rule 1: Each nurse is required to work one shift per day.

Rule 2: Each nurse gets at least one D/O shift per week.

Rule 3: Every nurse is entitled to three (3) days off after a night shift.

Rule 4: Every night shift is taken continuous for four conservative days.

Rule 5: The minimum number of nurses for morning shift should not be less than three. That is $M \geq 3$.

Rule 6: The number of nurses for both Afternoon and Night shifts should be at least two. That is $A \geq 2$ and N

Rule 7: Only the Principal Midwifery Officer (PMO) is entitled to Holiday off duty.

Rule 8: Principal midwifery officer is scheduled for only morning shifts and has day off duties on both Saturdays and Sundays.

The soft rules should be satisfied as much as possible but they can be violated. Since these preferences rules are mainly nurses choice on the timetable (duty roaster), they define the goodness of the roaster generated. The more preference planning rules a timetable respects, the better the timetable (roaster).

Rule 9: N shift are unwanted. Nurses should take turn to be assigned on N shift. To be exact, if there are Y nurses of a particular rank to be scheduled for N shift, each nurse in that group should have an N shift every Y days.

Rule 10: Immediately after an N shift, a nurse prefers to take on D/O shift. If it is not possible, a nurse prefers to take A shift. If it is possible, a nurse accepts an M shift.

Rule 11: Immediately after a VL shift, a nurse starts to work on a Monday. The nurse prefers to work in A shift on Monday. M shift on Tuesday and N shift on Wednesday.

Rule 12: If possible a nurse prefers to have consecutive holidays.

Rule 13: Each nurse should have equal opportunity to get D/O shift.

Rule 14: A nurse prefers to work with the following patterns.

Preferred Consecutive Patterns
M→D/O→A
M→N→A

Table 1.1

1.3 OBJECTIVES

It is a big problem in hospitals in Ghana to create schedules (duty roaster) for health staff (nurses), which do not victimize nurses in terms of both soft and hard constraints rules. Manually created programs can not deal with these problems despite the great effort required to form a duty roaster (timetable).

The purpose of this thesis is to develop an efficient and reliable solution to all these problems. In this thesis, we seek to construct a satisfying constraint-based nurse roastering system for the maternity ward of Ejura District Hospital in the Ashanti Region of Ghana. Eliminating most conflicting soft constraint is also a sole aim of this work.

1.4 JUSTIFICATION

The traditional nurse duty roaster (nurse schedule) construction for the maternity ward of Ejura District Hospital has over the year's generated problems with the roaster planner and nurses alike with the shifting methods of assignment. The mode though not egregious has many lapses, some of which are outlined below:

Conflict: There is the problem of conflicts in the existing method of nurse scheduling and this can be said to be very embarrassing to both the maternity ward and the Ejura District Hospital as a whole.

Time: Due to the number of hard hospital rules and soft nurse preference rules to be obeyed, it is evident that the construction of schedules for nurses is really a time consuming and most of the time ended up with lots of anomalies.

The manual time takes not less than two to four days to construct with a not-so-accurate result. The time involved in the manually inefficient timetable (duty roaster) construction could have been channelled into other departments of the Hospital.

Irregularities: Due to the problem of the existence of initial nurse – nurse conflict (who to go for night shift), the finalised roaster comes out with lots of conflictions.

Although there are relatively very few nurses, most of the hard rules are very tight and many soft planning rules are incompatible with each other. The duty planner allows nurses to request pre-assignments. The request often makes the roaster generation process more difficult. For example, suppose a group of nurse's request to have a day-off on Sunday. If this group is too large, it may over-reduce the available manpower to work on Sunday. It may also disturb normal sequences of some wanted (or unwanted) shifts and affect the fairness and quality of the roaster.

The system to be designed will primarily perform the most important task to eliminate conflicts by carefully studying the soft preference of nurses and to consider whose shift request to reject.

Time: Instead of solving a very large problem over a long time horizon, we seek to solve a sequence of smaller problems associated with shorter periods. With this, the time in constructing the roaster is reduced considerably.

Efficiency: Once the basic problem of conflict of interest with respect to shifts assignment is eliminated and computation is made on variably small number of nurses in a short period of time, the efficiency of making the schedule is appreciably higher compared to the former, though hundred percent efficiency achievements is impossible.

1.5 METHODOLOGY

Funding a suitable schedule (duty roaster) is not as simple as obtaining the best graph colouring algorithm. The algorithm may form the basis for a system, assigning nurses to schedules but there are many other points to consider. One major problem is how to allocate the nurses to various shifts. Grouping a set of nurses with a set of shifts is a problem, which must be solved for each time period. One common requirement is to minimise the number of consecutive or same day shift for a nurse to work.

We present the methods behind a spreadsheet type system motivated by the need for automatic assistance for nurse roastering (nurse scheduling). However, the system is designed to be a general purpose roastering spreadsheet. We create roastering, based on graph colouring, which will find times and shifts for each nurse. The method is the basis of a spreadsheet system. It uses a heuristic algorithm to find a series of almost maximal independent colour sets from the vertices of the graph, which are then assigned in turn to shifts using an algorithm with both hard and soft constraints.

We also use and study the structural properties of conflict graph instances that arise from nurse roastering problems and is based on the effectiveness of variety of graph colouring approaches. We seek intelligently ordered and intelligently – searched sequential colouring methods, as ~~well as an~~ integer and constraint programming formulations of graph colouring to arrest the problem of this work.

1.6 SCOPE AND LIMITATIONS

As previously discussed, this thesis seeks to study graph colouring with its associated applications to a real world problem. We proceed to the prominent one of all the applications. That is scheduling.

Specifically, we consider nurse scheduling system focussing primarily on the shift structure for the Ejura District Hospital. We limit our study to nurse roastering problem only and also consider a maternity ward of a district hospital with the hope that the work can be replicated to other district, municipal, general hospitals in Ghana. There is more room for using this work as basis or reference for further studies to investigate other schedule problems in other sectors of our economy.

1.7 ORGANISATION

This work is structured in a well planned manner. Chapter one gives a brief history of graph colouring and investigates the various practical applications of graph colouring. It discusses the forms of timetabling problems and tries to elaborate more on nurse scheduling system. Chapter two gives brief history on what others have studied around this subject area. Chapter three presents an overview of graph colouring. Basic terms are defined and explained. Theories and lemmas to help in our study are presented and proved. The nurse schedule model is discussed extensively in chapter four. Chapter five studies the results obtained and the necessary conclusion is made. Recommendation is seen in the chapter six.

CHAPTER TWO

2.1 LITRETURE REVIEW

Valls et al., (1996) examined a heterogeneous workforce assignment problem where the minimum number of workers needed to perform a machine load plan was calculated. The problem was represented as a restricted vertex colouring problem and a branch and bound algorithm was shown. The special characteristics of the graph to be coloured enable an efficient application of the branch and bound. Computational results indicated that the algorithm could solve problems of 50 tasks, 5, 10 and 15 machines and between 2 to 15 different kinds of workers in a few seconds.

Juhos et al., (2004) described a novel representation and ordering model that, aided by an evolutionary algorithm, was used in solving the graph k-colouring problem. Its strength lies in reducing the number of neighbours that need to be checked for validity. An empirical comparison was made with two other algorithms on a popular selection of problem instances and on a suite of instances in the phase transition. The new representation in combination with a heuristic mutation operator showed promising results.

Bouchand et al., (1986) studied numerically the (planar) graph colouring problem with q colours. For $q = 4$, when a perfect colouring could be achieved, the solutions were scattered "randomly" (as far as triangle correlations are concerned) in configuration space. On the contrary, for $q = 3$, colouring was always imperfect, but the optional solutions seem to organize themselves in an ultrametric way. This illustrated rather well the role of *frustration* on the configuration space landscape. The authors discussed the importance of the distance chosen.

Culberson and Gent (2001) denned the 'frozen development' of colouring random graphs and identified two nodes in a graph as frozen if they were the same colour in all legal colourings. This was analogous to studies of the development of a backbone or spine in SAT (the Satis ability problem). The authors described in detail the algorithmic techniques used to study frozen development and presented strong empirical evidence that freezing in 3-colouring is sudden. A single edge typically caused the size of the graph to collapse in size by 28% and used the frozen development to calculate unbiased estimates of probability of colourability in random graphs, even where this probability was low. The links between frozen development and the solution cost of graph colouring was investigated. In SAT, a discontinuity in the order parameter was correlated with the hardness of SAT instances; data for colouring was suggestive of an asymptotic discontinuity. The uncolourability threshold was known to give rise to hard test instances for graph-colouring. Evidence that the cost of colouring threshold graphs grows exponentially, when using either a specialist colouring program, or encoding into SAT, or even when using the best of both techniques were presented. Theoretical and empirical evidence showed that the size of the smallest uncolourable sub graphs of threshold graphs became large as the number of nodes in graphs increases. The application of their work to the statistical mechanics analysis of colouring was discussed extensively.

The graph-theoretic parameter that has probably received the most attention over the years in the chromatic number. As is well-known, the colouring problem is an NP-

Complete problem. Lie et al., (2002) solved by means of molecular biology techniques to this effect. The algorithm was highly parallel and had satisfactory fidelity. Their work showed further evidence for the ability of DNA computing to solve NP-Complete problems. (Graph colouring problem).

The ever growing number of wireless communications systems deployed around the globe have made the optimal assignment of a limited radio frequency spectrum a problem of primary importance, at issue are planning models for permanent spectrum allocation, licensing, regulation, and network design. Further at issue are on-line algorithms for dynamically assigning frequencies to users within an established network. Applications include aeronautical mobile, land mobile, maritime mobile, broadcast, land fixed (pointto-point), and satellite systems.

Murphey et al., (1999) surveyed researches conducted by theoreticians, engineers, and computer scientists regarding the frequency assignment problem (FAP) in all of its guises. Their paper began by defining some to the more common types of FAPs. It continued with a discussion on measures of optimality relating to the use of spectrum, models of interference, and mathematical representations of many FAPs, both in graph theoretic terms, and as mathematical programs. Graph theory and, in particular, graph colouring play an important role in the FAP since, in many instances, the FAP in cast in a form which closely resembles a graph colouring. Theoretical results that bound optimal solutions for special FAP structures were presented. Exact algorithms for general FAPs were explained, and since many FAP instances are computationally hard, much space was devoted to approximate algorithms. Their paper concluded with a review of evaluation methods for FAP algorithms, test problem generators, and a discussion of the underlying engineering issues that was considered when generating test problem.

Hedetniemi (2002) proposed two new self-stabilizing distributed algorithms for proper ¡ (¡ is the maximum degree of node in the graph) colouring of arbitrary system graphs. Both algorithms were capable of working with multiple types of demons (schedulers).

The first algorithm converges in Ç Ñ moves while the second converges in at most Ò moves (Ò is the number of nodes and Ñ is the number of edges in the graph). The second improvement was that neither of the proposed algorithms requires each node to have knowledge of ¡. Further, the colouring produced by their first algorithm provided an interesting special case of colouring, e.g., Grundy Colouring.

The problem of properly colouring the vertices (or edges) of a graph using for each vertex (or edge) a colour from a prescribed list of permissible colours, received a considerable amount of attention. Alon (1993) described the techniques applied in his study of this subject, which combined combinatorial, algebraic and probabilistic methods, and discussed several intriguing conjectures and open problems. This was mainly a survey of recent and less recent results in the area, but it contained several new results as well.

Simulated annealing is also a very successful heuristic for various problems in combinatorial optimization. An application of simulated annealing to the 3-colouring problem was considered by Nolte and Schrader (1999). In contrast to many good empirical results they ~~showed for~~ a certain class of graphs that the expected first hitting time of a proper colouring, given an arbitrary cooling scheme, was of an exponential size and proved the convergence of simulated annealing to an optimal solution in an exponential time.

Fre et al., (2006) were interested in the graph colouring problem. They proposed an exact method based on a linear-decomposition of the graph. The complexity of this method was exponential according to the linear width of the entry graph, but linear according to its number of vertices. Presentations of some experiments were performed on literature instances, and their method was useful to solve more quickly than other exact algorithms instances with small linear width.

D'Hondt (2008) investigated quantum algorithms for graph colouring problems, in particular for 2- and 3-colouring of graphs. The main goal was to establish a set of quantum representations and operations suitable for the problem at hand and proposed a unitary-as well as measurement-based quantum computations, also taking inspiration from answer set programming, a form of declarative programming close to traditional logic programming. The approach used was one in which he first generate arbitrary solutions to the problem, then constraining those according to the problem's input. Though he did not achieve fundamental speed-ups, his algorithms showed show quantum concepts could be used for programming and moreover exhibit structural differences. For example, the computation of all possible colourings at the same time. Comparing, his algorithms with classical ones, highlighting how the same type of difficulties gave rise to NP-complete behaviour, and proposed possible improvements.

Graph-colouring register allocation is an elegant and extremely popular optimization for modern machines. But as currently formulated, it does not handle two characteristics commonly found in commercial architectures. First, a single name may appear in multiple register classes, where a class is set of register names that are interchangeable in a particular role. Second multiple register names may be aliases for a single hardware register. Holloway et al., (1993) presented a generalization of graph-colouring register allocation that handle these problematic characteristics while preserving the elegance and practicality of traditional graph colouring. Their generalization adapts easily to a new target machines, requiring only the sets of names in the register classes and a map of the register aliases. It also drops easily into a well-known graph-colouring allocator, is efficient at compile time, and produces high-quality code.

Duffy et al., (2006) analysed the complexity of decentralised colouring algorithm that had recently been proposed for channel selection in wireless computer networks. Colouring a graph with its chromatic number of colours is known to be NP-hard. Identify an algorithm in which decisions are made locally with no information about the graph's global structure is particularly challenging.

Koyuneu and Secir (2004) used graph colouring algorithm to generate the student weekly time table in a typical university department. Their problem was a Hode-point problem and it could not be solved in the polynomial domain. Various constraints in weekly scheduling such as lecturer demands, course hours and laboratory allocations were confronted and weekly time tables were generated for first, second, third and fourth year students in a typical semester.

Burke et al., (1995) developed a general system able to cope with the ever changing requirements of large educational institutions. They presented the methods and techniques behind such a system. Graph Colouring and room allocation algorithms were also presented and it was shown in their basis of a flexible and widely applicable timetabling system.

The intended to overcome the problem of intractability by producing spreadsheet type system that the user could be guided in an informed and useful way. That gives the user control of the search and the possibility of backtracking where no reasonable solution is found, while still letting the heuristic algorithms to the hard work. Their approach cannot guarantee an optimal solution but it can guarantee a solution the user is happy with.

CHAPTER THREE

AN OVERVIEW OF GRAPH THEORY AND GRAPH COLOURING

3.1 AN INTRODUCTION

In this chapter, we elaborate in depth the theory of graphs in general and twist our concentration to graph colouring; definitions of terminologies and its applications-graph theory is an old subject with many modern applications. Its basic ideas were introduced in the eighteenth century by the great Swiss mathematician Leonhard Euler. [14]. He used graphs to solve the famous Kongsberg bridge problem. It is notably known that problems in almost every conceivable discipline can be solved using graph models. Graph colouring is a special case of graph labelling. It is an assignment of labels traditionally called "colours" to elements of a graph subject to certain constraints. In general, considering a graph G, a vertex colouring or simply a colouring of G is an assignment of colours to the vertices of G such that adjacent vertices have different colours. We say that G is $n-$*colourable* if there exist a colouring of G which uses n colours.

3.2 GRAPH TERMINOLOGY

3.2.1 Types of graphs

A graph is a mathematical object composed of points known as vertices or nodes and lines connecting them known as edges.

Symbolically, we write a graph as $G=(V,E)$ where V is the set of vertices and E is the set of pairs representing the edges.

A simple graph $G=(V,E)$ consists of V called edges. For example considering a simple graph

In figure 2.1 where the graph $G=(V,E)$ has;

V consisting of vertices A, B, C, D and V= {A, B, C, D}

E consisting of edges e_1= {A, B}, e_2= {B, C}, e_3={C, D},

e_4= {A, C}, e_5= {B, D}

Therefore E={(A,B),(B,C),(C,D),(A,C),(B,D)}

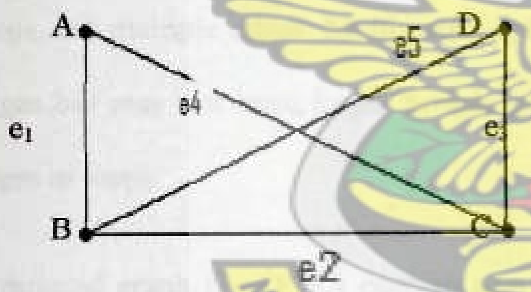
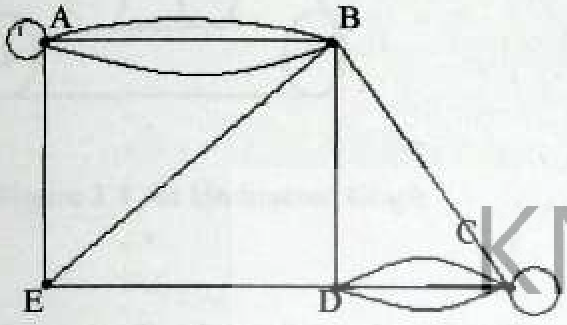


Figure 3.1 A simple graph

A multigraph $G=(V,E)$ consists of a set of vertices, a set E of edges and a function f from E to $\{(u,v)\mid u,v\in V,u\neq v\}$.The edges e_1 and e_2 are called multiple or parallel edges if $f(e_1)=f(e_2)$

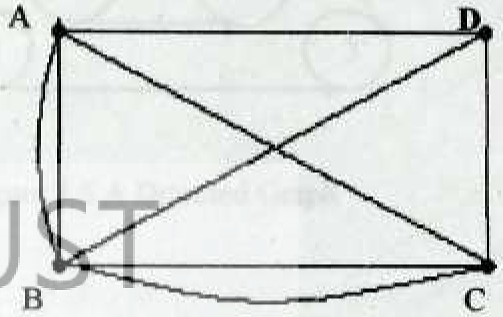
Occasionally, a computer network may contain a telephone line from a computer to itself for diagnostic purposes. The network resulted from this is called pseudograph.

A pseudograph $G=(V,E)$ consists of a set V of vertices, a set E of edges and a function f from E to $\{\{u,v\}|u,v\in V\}$. An edge is a loop if $f(e)$ = {u, u} = {u} for some $u\in V$



A Pseudograph

Figure 3.3



A Multigraph

Figure 3.2

Pseudo graphs are the most general type of undirected graphs since they may contain loops and multiple edges. Multigraphs are undirected graphs that may contain multiple edges but may not have loops. Simple graphs are undirected graphs with no multiple edges or loops.

A directed graph $G=(V,E)$ consists of a set of vertices V and a set of edges E that are ordered pairs of elements of V.

A directed graph is often called digraph where (u, v) represents a directed edge from U to V. An undirected graph results when the edge (u, v) is the same as (u, v).

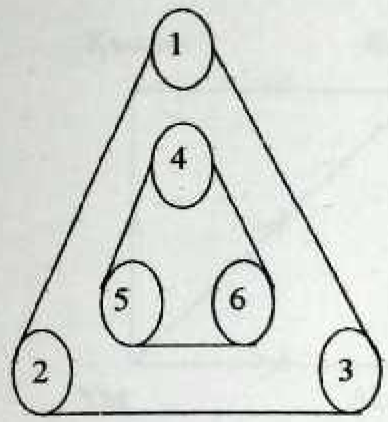


Figure 3.4 An Undirected Graph

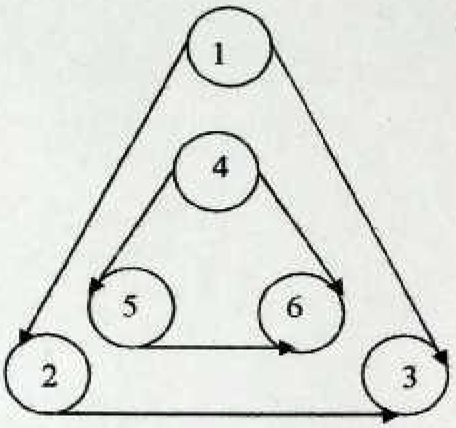


Figure 3.5 A Directed Graph

Symbolically, the undirected graph will be represented as

$G=(V,E)$

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

$E=\{(1,2),(3,1),(3,2),(4,5),(6,4),(5,6)\}$

While the Directed graph is

$G=(V,E)$

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

$E=\{(1,2),(1,3),(2,3),(4,5),(4,6),(5,6)\}$

An influence Graph is a directed graph used to model the behaviour of a group of people in which certain persons can influence the thinking of other. Each person of the group is represented by a vertex. There is a directed edge from vertex α to vertex β when the person represented by the vertex α influences the person represented by vertex β.

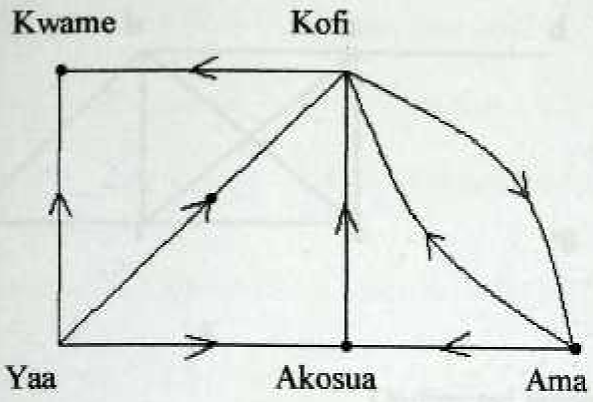


Figure 3.6 An influence Graph

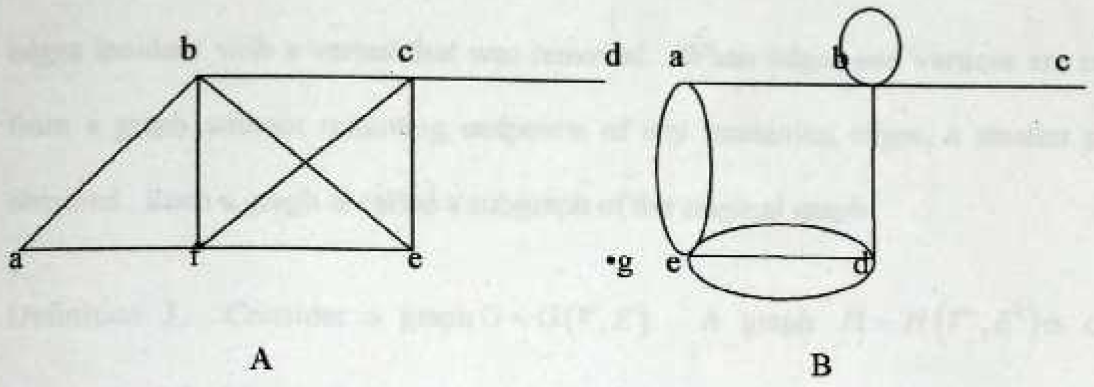
From the influence graph we get to know that Yaa can or will influence Kwame, Kofi and Akosua but no one can influence her. Also Ama and Kofi can influence each other.

3.2.2 Basic Definitions

Definition 1. Two Vertices U and V in an undirected graph G are called adjacent (or neighbours) in G if $\{u, v\}$ is an edge of G. If e $>\{u, v\}$, the edge e is called incident with the vertices u and v. The edge e is also said to connect u and v. The vertices u and v are called endpoints of the edge $\{u, v\}$ .Investigating how many edges are incident to a vertex, we consider the second definition.

Definition 2. The degree of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex V is denoted by $\deg(v)$.

We present the following illustrations to elaborate more on the two definitions. We hope that it will explain the definitions much clearly even to the layman. Considering the undirected graphs A and B, the degrees of the vertices in the graphs are as follows;



Undirected graphs A and B

Figure 3.7

In graph A, there are seven vertices and each vertex has a degree:

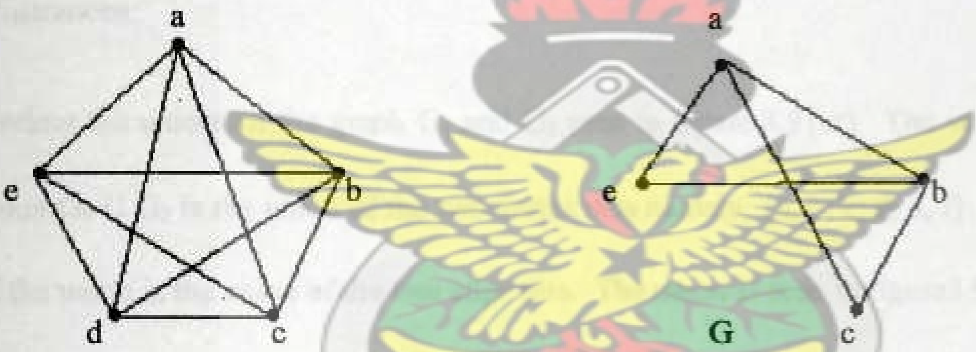
$\deg(a) = 2$, since only two edges ({a, b} and {a, f}) are incident with vertex a.

Similarly deg (b) = deg (c) = deg (f) = 4 because four edges incident with each vertex. deg (d) = 1, deg (e) and deg (g) = 0 since it has no edge(s) incident with it. With graph B, there are five vertices. deg (a) = 4, deg (c) = 1, deg (d) = 5 and deg (e) = deg (b) = 6. It is noted that the loop at vertex b constitute degree of two for vertex b. A vertex of degree zero (0) is called isolated. It follow from our illustration that an isolated vertex is not adjacent to any vertex as it is the case for vertex g in graph A.

A vertex is pendant if and only if it has degree of one (1). Consequently, a pendant vertex is adjacent to exactly one other vertex. Vertex d in graph A is pendant. The vertex c in graph B is also pendant. Sometimes only part of a graph is needed to solve a particular problem. In the graph model for large network, the vertices corresponding to computer centers can be removed other than some of interest. And we can remove all edges incident with a vertex that was removed. When edges and vertices are removed from a graph without removing endpoints of any remaining edges, a smaller graph is obtained. Such a graph is called a subgraph of the original graph.

Definition 3. Consider a graph $G = G(V, E)$. A graph $H = H\left(V^1, E^1\right)$ is called a subgraph of G if the vertices and edges of H are contained in the vertices and edges of G, that is, if $V^1 \subseteq V$ and $E^1 \subseteq E$.

The graph G is a subgraph of K_5



K_5

A Subgraph of K_5

Figure 3.8

Two or more graphs can be combined in many ways. The new graph that contains all the vertices and edges of these graphs is called the union of graphs.

Definition 4. The union of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$. The union of G_1 and G_2 is denoted by $G_1 \cup G_2$.

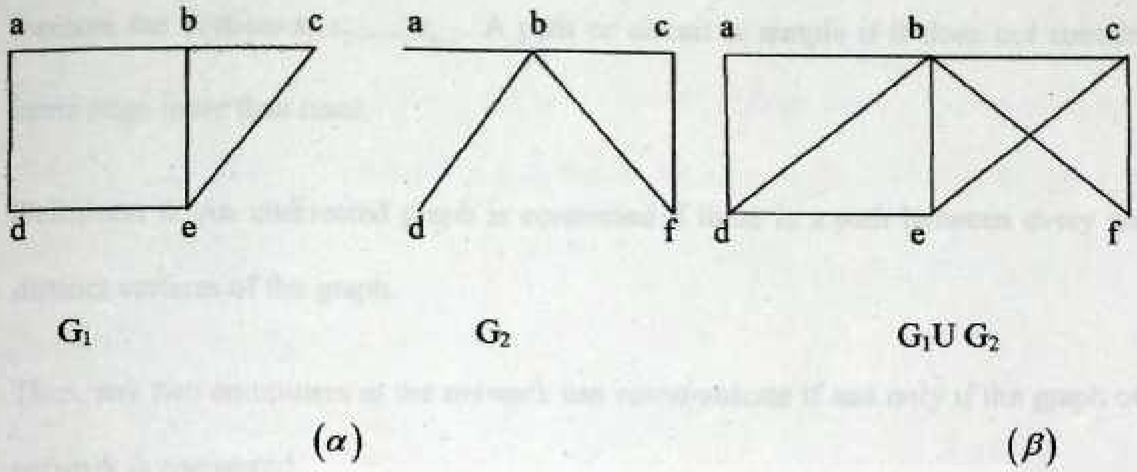


Figure 3.9

Figure 3.9(α) The simple Graphs G₁ and G₂ Figure 3.9(β) Their union G₁

Illustrations:

Finding the unions of the graph G₁ and G₂ seen in figure 3.9(α) The vertex set of the union G₁ U G₂ is the union of the two vertex sets namely, {a, b, c, d, e, f}. The edge set of the union is the union of the two edge sets. The union is seen in figure3.9(β).

Definition 5. A path of length n from u to v, where n is a positive integer, in an undirected graph is a sequence of edged $e_1,\ldots,e_n$ of the graph such that $f(e_1)=\{x_0,x_1\}, f(e_2)=\{x_1,x_2\},\ldots,f(e_n)=\{x_{n-1},x_n\}$, where $x_0=u$ and $x_n=v$. when the graph is simple, we denote this path by its vertex sequence $x_0,x_1,\ldots,x_n$ (since listing these vertices uniquely determines the path). The path or circuit is a circuit if it begins and ends at the same vertex, that is if $u=v$. The path or circuit is said to pass through a traverse the vertices $x_1, x_2, \ldots, x_{n-1}$. A path or circuit is simple if it does not contain the same edge more than once.

Definition 6. An undirected graph is connected if there is a path between every pair of distinct vertices of the graph.

Thus, any two computers in the network can communicate if and only if the graph of this network is connected.

The graph G in figure 3.10 is connected, since for every pair of distinct vertices there is a path between them. However the graph H in figure 3.10 is not connected. For instance, there is no path in H between a and d.

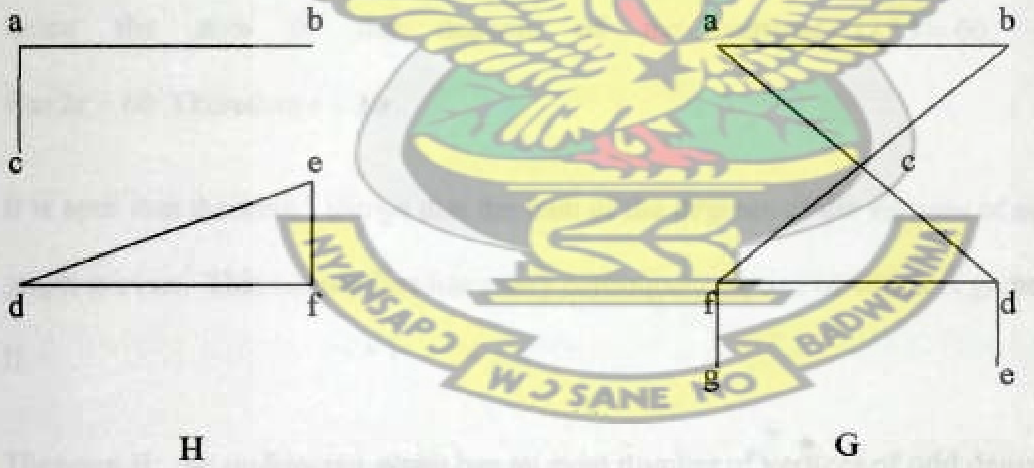


Figure 3.10

3.2.3 Theorems

Each edge of a graph contributes two (2) to the sum of the degrees of the vertices since and edge is incident with exactly two (possibly equal) vertices. This means that the sum of the degrees of the vertices is twice the number of edges.

Theorem I: The Handshaking Theorem:

Let $G=(V,E)$ be an undirected graph with edges then

$$2e=\sum_{v\in V}\deg(v)$$

The Handshaking theorem applies even if multiple edges and loop are present in an undirected graph.

Example, how many edges are there in a graph with 12 vertices each of degree 5?

Since the sum of the degrees of the vertices $=12*5=60$, it follows that $2e=60$ .Therefore $e=30$.

It is seen that theorem I shows that the sum of the degrees of the vertices of an undirected graph is even. This simple fact has many consequences, one of which is given in theorem II.

Theorem II: An undirected graph has an even number of vertices of odd degrees.

Proof: Let V_1 *and* V_2 be the set of vertices of even degree and the set of vertices of odd degree respectively, in an undirected graph $G=(V,E)$ then

$$2e = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v)$$

Since $\deg(v)$ is even for $v \in V_1$, the first term in the right hand side of the last equality is even. Furthermore, the sum of the two terms on the right hand side of the last equality is even, since this sum is 2e. Hence the second term in the same is also even. Since all the terms in this sum are odd, there must be an even number of such terms. Thus, there are even numbers of vertices of odd degrees. Hence the proof.

Theorem III: EULER'S FORMULA: Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G.

Then $\quad r = e - v = 2$.

Proof: First, we specify a planar representation of G. We will prove the theorem by constructing a sequence of subgraphs $G_1, G_2, G_3, \ldots, G_e = G$ successively adding an edge at each stage. This is done using the following inductive definition.

Arbitrarily picking one edge of G to obtain G_1. Obtain G_n from G_{n-1} by arbitrarily adding an edge that is incident with a vertex already in G_{n-1}, adding the other vertex incident with a vertex already in G_{n-1}. This construction is possible since G is connected. G is obtained after e edges are added. Let r_n, e_n and v_n represent the number of regions edges and vertices of the planar representation of G_n induced by the planar representation of G respectively.

The proof proceeds by induction. The relationship $r_1 = e_1 - v_1 + 2$ is true for G_1 since $e_1 = 1$, $v_1 = 2$ and $r_1 = 1$. This is shown in figure 3.11.

Now assume that $r_n = e_n - v_n + 2$. Let $\{a_{n+1}, b_{n+1}\}$ be the edge that is added to G_n to obtain G_{n+1}. There are two possibilities to consider. In the first case, both a_{n+1} and b_{n-1} are already in G_n. These two vertices must be on the boundary of a common region R or else it would be impossible to add the edge $\{a_{n+1}, b_{n+1}\}$ to G_n without two edges crossing (and a_{n+1} is planar). The addition of this new edge splits R into two regions. Consequently, in this case, $r_{n+1} = r_n + 1$, $e_{n+1} = e_n + 1$ and $v_{n+1} = v_n$. Thus, each side of the formula relating the number of regions, edges and vertices increases by exactly one, so this formula is true. In other words $r_{n+1} = e_{n+1} - v_{n+1} + 2$. This case is illustrated in fig $3.12(\alpha)$

In the second case one of the two vertices of the new edge is not already in G_n. Suppose that a_{n+1} is in G_n but that b_{n+1} is not. Adding this new edge does not produce any new regions, since b_{n+1} must be in a region that has a_{n+1} on its boundary. Consequently, $r_{n+1} = r_n$. Moreover, $e_{n+1} = e_n + 1$ and $v_{n+1} = v_n + 1$. Each side of the formula relating the number of regions, edges and vertices remains the same, so the formula is still true. In other words, $r_{n+1} = e_{n+1} - v_{n+1} + 2$. This case is illustrated in fig 3.12(β). Hence from the induction argument, $r_n = e_n - v_n + 2$ for all n. Since the original graph is the graph G_e obtained after e edges have been added, the theorem is true. Hence the proof.

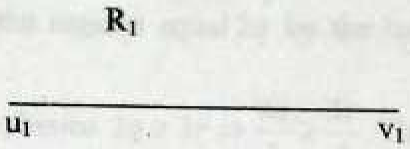
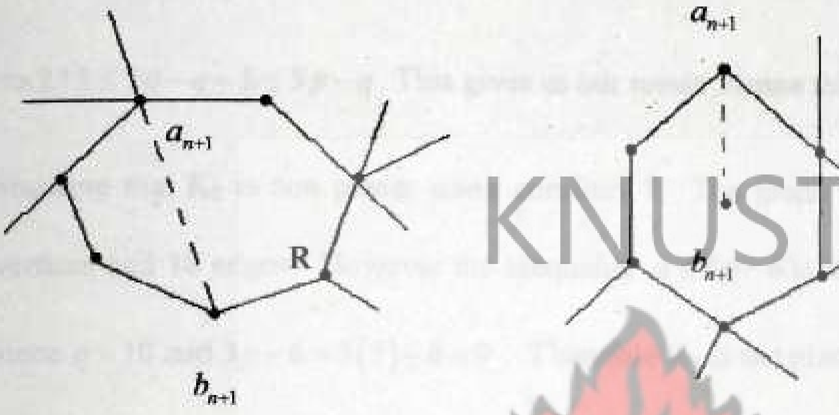


Figure 3.11. The basis case of the proof of Euler's formula.



(α) Figure 3.12 (β)

Adding an edge to G_n to produce G_{n+1}.

Illustration for Euler's formula: Finding the number of regions does a representation of a planar graph with 20 vertices, each of degree 3 split the plane. We know that the graph has 20 vertices, each of degree 3, so $v = 20$. The sum of the degree of the vertices, $3v = 3*20 = 60$,is equal to twice the number of edges $2e$, this implies $2e = 60 \Rightarrow e = 30$ Consequently from Euler's formula, the number of regions is $r = e - v + 2 = 30 - 20 + 2 = 12$.

Corollary I: Let G be a connected planar graph with p vertices and q edges, where $p \geq 3$. Then $q \leq 3p - 6$

Proof: Let r be the number of regions in a planar representation of G. By Euler's formula $p - q + r = 2$

Now the sum of the degree of the regions equal $2q$ by the handshaking theorem. But each region has degree 3 or more, hence $2q \ge 3r \Rightarrow \frac{2q}{3} \ge \frac{3r}{3}$. Thus r $\le \frac{2q}{3}$.

Substituting this in Euler's formula gives $2 = p-q+r \le p-q+\frac{2q}{3}$ or $2 \le p-\frac{q}{3}$

$\Rightarrow 2*3 \le 3p-q = 6 \le 3p-q$. This gives us our result. Hence the proof.

Showing that K_5 is non planar using corollary I: The graph K_5 (figure 3.13) has five vertices and 10 edges. However the inequality $q \le 3p-6$ is not satisfied for this graph since $q=10$ and $3p-6=3(5)-6=9$. Therefore K_5 is not planar.

3.2.4 Special graphs

Finite Graph: A multigraph is said to finite if it has a finite number of vertices and a finite number of edges. Observe that a graph with a finite number of vertices must automatically have a finite number of edges and so must be finite.

Trivial Graph: The finite graph with one vertex and no edges (that is a single point) is called the trivial graph.

Complete Graph: A graph G is said to be complete if every vertex in G is connected to every other vertex in G. The complete graph on n vertices denoted by K_n, is the simple graph that contains exactly one edge between each pair of distinct vertices. Figure 3.13 shows the graph K_1 through K_6

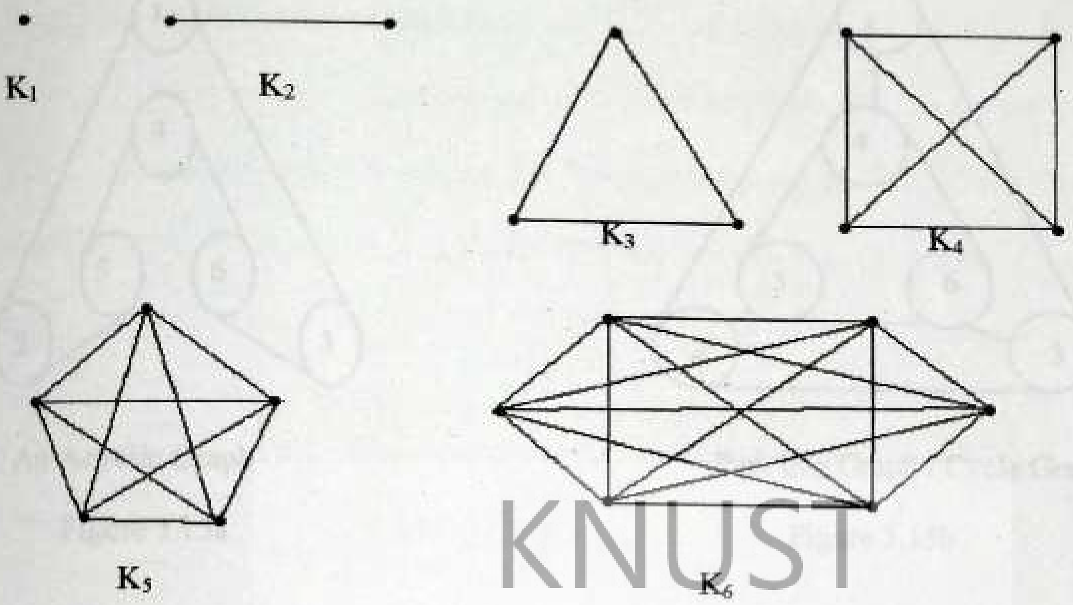
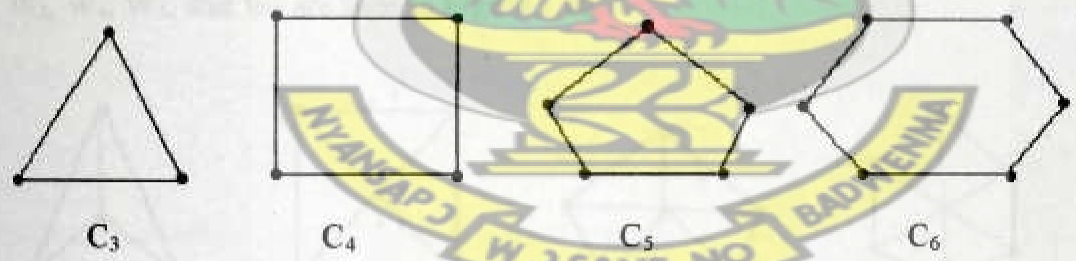


Figure 3.13 The Graphs of K$_n$, $1 \le n \le 6$

Cycles: The Cycle $C_n, n \geq 3$, consists of n vertices $V_1, V_2, V_3, \ldots, V_n$ and edges $\{V_1, V_2\}, \{V_2, V_3\}, \ldots, \{V_{n-1}, V_n\}$ and $\{V_n, V_1\}$.

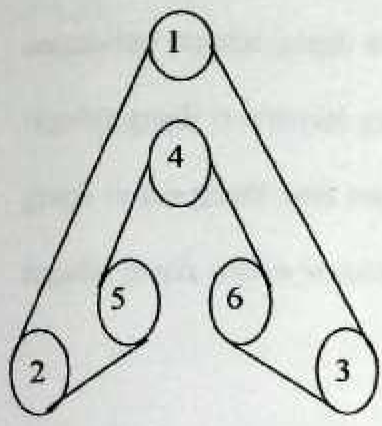
Examples of cycles C$_3$, C$_4$, C$_5$, and C$_6$ are displayed below.



The Cycles C$_n$, $3 \le n \le 6$

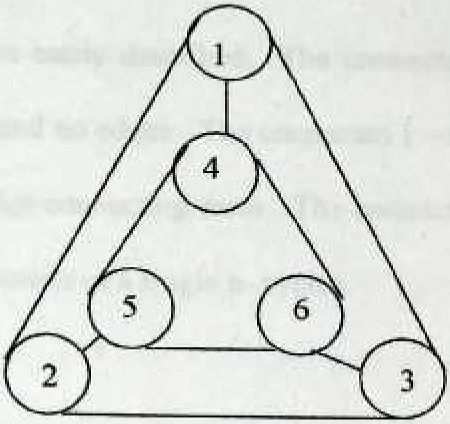
Figure 3.14

In short a cycle is a path where the first and last vertices are the same. A graph containing no cycles is said to be acyclic. In the graph below figure 3.15b, the path 1, 2, 3, 1 is a cycle.



An Acyclic Graph

Figure 3.15a



Path in a Graph / Cycle Graph

Figure 3.15b

An acyclic undirected graph is called a Forest. An acyclic connected undirected graph is a Tree. The graph above (Figure 3.15a) can be considered as an acyclic, a forest and also a tree.

Wheels: We obtain the wheel W_n when we add an additional vertex to the cycle C_n, for n ≥ 3, and connect this new vertex to each of the n vertices in C_n by new edges. The wheels W_3, W_4, W_5, and W_6 are seen in Figure 3.16

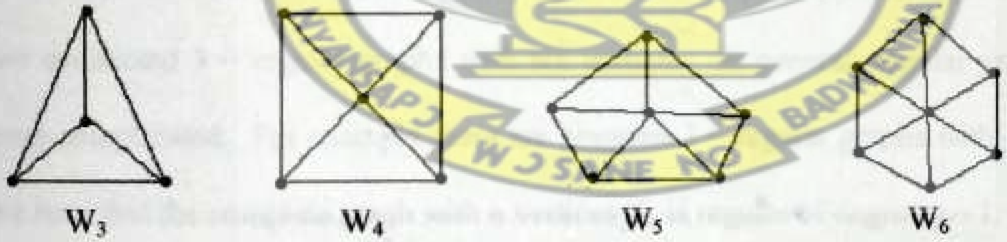
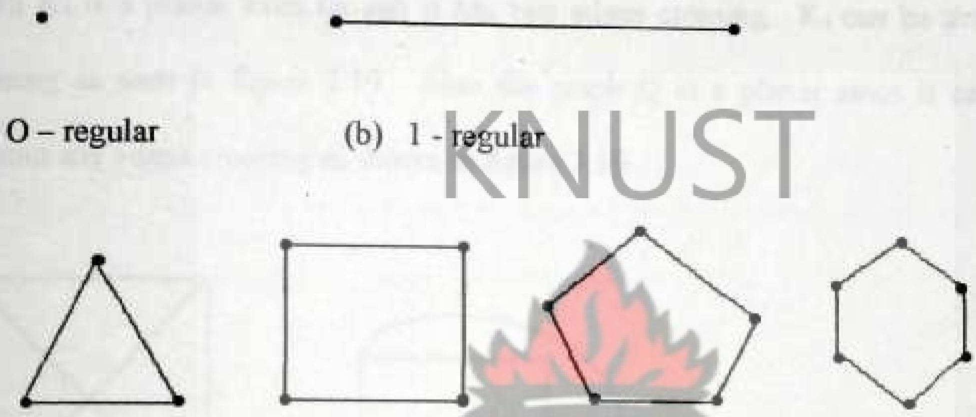


Figure 3.16 The Wheels W_n, $3 \leq n \leq 6$

Regular Graphs: A graph G is regular of degree K or K-regular if every vertex has degree K. In other words, a graph is regular if every vertex has the same degree. The connected regular graph of degrees 0, 1or 2 are easily described. The connected O – regular graph is a trivial graph with one vertex and no edges. The connected I – regular graph is the graph with two vertices and one edge connecting them. The connected 2 – regular graph with n vertices is a graph which consists of a single n- cycles.



(a) O – regular (b) 1 - regular

(c) 2 – regular

Figure 3.17 Regular graphs

The 3 – regular graph must have an even number of vertices since the sum of the degrees of the vertices is an even number (Theorem I: Handshaking theorem) Figure 3.18 shows two connected 3 – regular graphs with six vertices. In general, regular graphs can be quite complicated. For example there are nineteen 3 – regular graphs with ten vertices. We note that the complete graph with n vertices K_n is regular of degree n – 1.

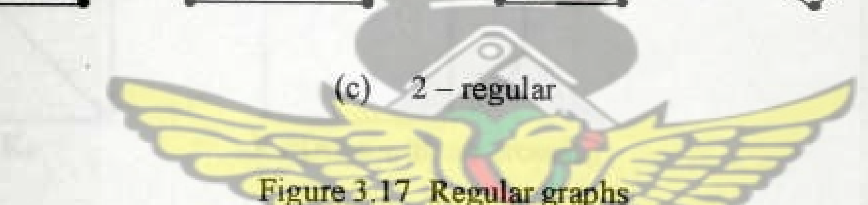
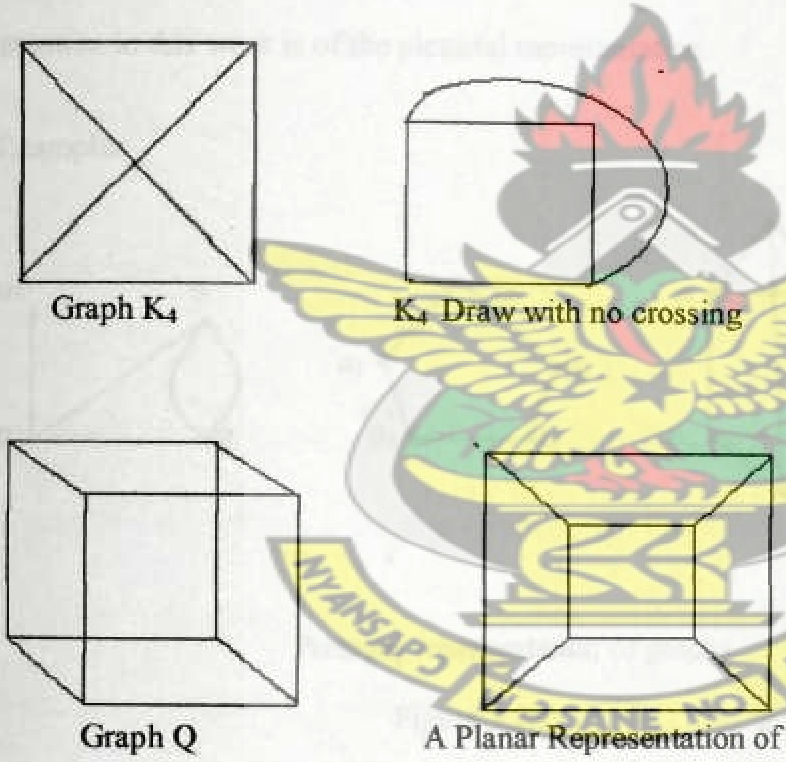


Figure 3.18 3 - regular

Planar graph: A graph is called planar if it can be drawn in the plane without any crossing (where a crossing of edges is the intersection of the lines or arcs representing them at a point other than their common end point). Such a drawing is called a planar representation of the graph. A graph may be planar even if it is usually drawn with crossings, since it may be possible to draw it in a different way without crossing. The graph K_4 is a planar even though it has two edges crossing. K_4 can be drawn without crossing as seen in figure 3.19. Also the graph Q is a planar since it can be drawn without any edges crossing as shown in figure 3.19



Graph K_4

K_4 Draw with no crossing

Graph Q

A Planar Representation of Graph Q

Figure 3.19

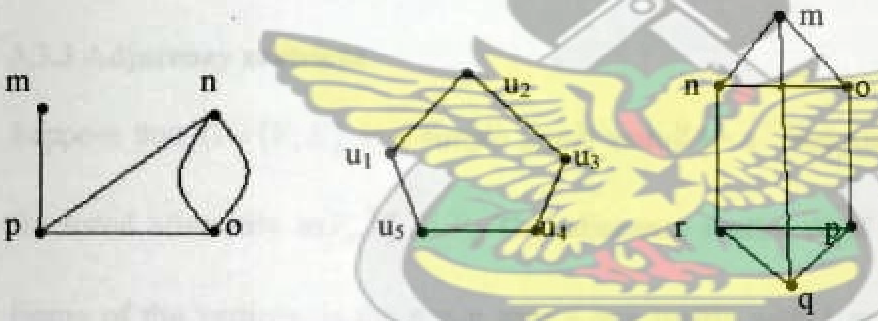
3.3 GRAPH REPRESENTATION

There are many useful ways to represent graphs. Graph representation can be state with the commonest are; the pictorial view point which we all are comfortable with and also used to. Then moves to the set description. The Adjacency matrices, incidence matrices through the Adjacency list computer.

3.3.1 Pictorial view of graphs

That is when graphs are represented in diagram form. In fact all the various graphs mention in this work is of the pictorial representation.

Examples



Pictorial representation of graphs

Figure 3.20

3.3.2 Set description

This is a mathematical representation of a graph. An example is a graph $G=(V,E)$ in figure 3.4 where $V=\{V_1,V_2,V_3,\ldots,V_n\}$ is the set of n vertices and $E=\{e_1,e_2,e_3,\ldots,e_m\}$ is the set of m edges where each edge $e_k=\{V_i,V_j\}$

So the graph's set description is

$G=(V,E)$

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

$E=\{(1,2),(3,1),(3,2),(4,5),(6,4),(5,6)\}$

3.3.3 Adjacency matrices

Suppose that $G=(V,E)$ is a simple graph were $|V|=n$ Suppose that the vertices of G are listed arbitrarily as $V_1,V_2,\ldots,V_n$. The adjacency matrix A of G, with respect to this listing of the vertices, is the n × n zero – one matrix with 1 as its $(i,j)th$ entry when V_i and V_j are adjacent and O as its $(i,j)th$ entry when they are not adjacent. In other words, if it's adjacency matrix is

$$A=\left[a_{ij}\right]\text{, then}\qquad a_{ij}=\begin{cases}1\text{ if }\{v_i,v_j\}\text{ is the edge of G}\\ 0\qquad\qquad \textit{Otherwise}\end{cases}$$

An adjacency matrix of a graph is based on the ordering chosen for the vertices. Hence there are as many as n! different adjacency matrices for a graph with n vertices since there are n! different orderings of n vertices. The adjacency matrix of a simple graph is symmetric, that is $a_{ij} = a_{ji}$ since both of these entries are 1 when V_i and V_j are adjacent and both are O otherwise. Since a simply graph has no loops each entry $a_{ii}, i = 1,2,3,\ldots,n$ is O. When there are relatively few edges in a graph, the adjacency matrix is a SPARSE MATRIX, which is a matrix with few non-zero entries.

Transforming the graph F into an adjacency matrix with the order vertices as a, b, c, d, we have

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Or

Vertices	A	b	C	d
A	0	1	1	1
B	1	0	1	0
C	1	1	0	0
d	1	0	0	0

For the Pseudograph W the transform adjacency matrix with vertices ordering u_1, u_2, u_3, and u_4 is

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

Or

Vertices	u_1	u_2	u_3	u_4
u_1	0	3	0	2
u_2	3	0	1	1
u_3	0	1	1	2
u_4	2	1	2	0

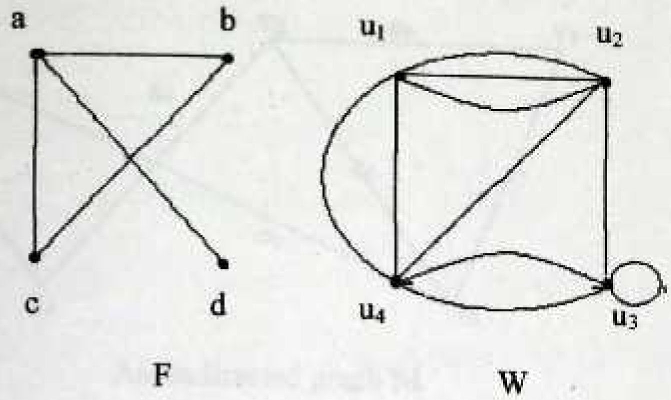


Figure 3.21

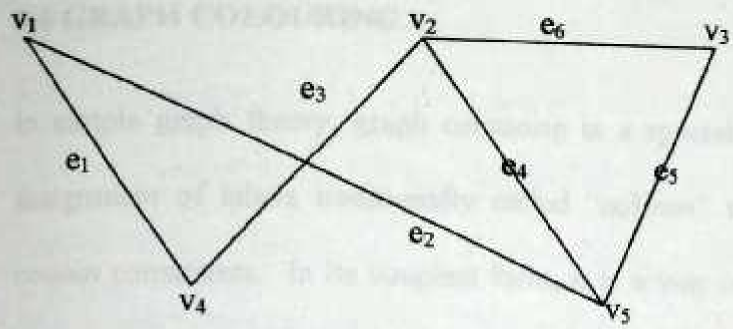
3.3.4 Incidence matrices

Let$G=(V,E)$ be an undirected graph. Suppose that$V_1,V_2,\ldots,V_n$. are the vertices and$e_1,e_2,\ldots,e_m$ are the edges of G. Then the incidence matrix with respect to this ordering of V and E is the n × m matrix $B=\left[b_{ij}\right]$, where

$$b_{ij}=\begin{cases}1 \text{ when edge } e_j \text{ is incident with } v_i\\ 0 \qquad\qquad\qquad \textit{Otherwise}\end{cases}$$

Transforming an undirected graph M with an incident matrix, we have

$$b_{ij}=\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ v_1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 1 & 1 & 0 & 1 \\ v_3 & 0 & 0 & 0 & 0 & 1 & 1 \\ v_4 & 1 & 0 & 1 & 0 & 0 & 0 \\ v_5 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}$$



An undirected graph M

Figure 3.22

3.3.5 Adjacency list (computer)

This is one –dimensional array of size |V| each element of which points to an array or a linked list of adjacent vertices. So the undirected graph in Figure 3.4 will have an adjacency list as

Vertex 1	→	{2, 3}
Vertex 2	→	{1, 3}
Vertex 3	→	{1, 2}
Vertex 4	→	{5, 6}
Vertex 5	→	{4, 6}
Vertex 6	→	{4, 5}

3.4 GRAPH COLOURING

In simple graph theory, graph colouring is a special case of graph labelling. It is an assignment of labels traditionally called "colours" to elements of a graph subject to certain constraints. In its simplest form, it is a way of colouring the vertices of a graph such that no two adjacent vertices share the same colour. This type of colouring is called Vertex Colouring. Similarly, an edge colouring assigns a colour to each edge so that no two adjacent edges share the same colour and a face colouring of a planar graph assigns colour to each face or region so that no two faces that share a boundary have the same colour.

A colour of a graph is always assumed to be a vertex colouring, namely an assignment of colours to the vertices of the graph.

Given a graph G, a vertex colouring of G is a function F, from the vertex of G to a set C whose elements are called colours such that no two vertices have the same colour. It is often both conventional and convenient to use number 1, 2, 3... for the colours.

A colouring using at most K colours is called a proper K-colouring. Hence a proper K – colouring of G is a colouring function which uses exactly K colours and satisfies the property that f (y) whenever x and y are adjacent in G. We say such a graph G is K colourable. If G has a loop, then G has no proper colouring since f (u) = f (y).

The chromatic number X (G) of G is the minimum K such that there exist a proper K colouring G. A clique Kr of G is an r-vertex sub graph of G in which each pair of vertices in Kr share an edge.

For example K_2 is a single edge and K_3 is a triangle. It is easy to see that the size of the maximum clique Kr of G gives a lower bound on the chromatic number of G since X (Kr) = r and therefore X (G) ≥ r. It is also easy to see that X (G) ≤ Δ (G) + 1 where Δ (G) denotes he maximum degree of G. A minimum colouring of G is a proper colouring that uses a few colours as possible. That is X (G) colours are both necessary and sufficient for a proper colouring of G. Many a times one is presented with a graph and must determine its chromatic number as well as produce a minimum colouring of the graph. This problem is known is graph colouring as the minimum colouring problem.

3.5 GRAPH COLOURING, AN INTERGER LINEAR PROGRAM

An integer program is a discrete optimization problem of the form.

Minimise (or Maximise) $F (x_1, x_2, ..., x_n)$

Subject to a set of m equality constraints

$$g_1 (x_1, x_2, . . ., x_n) = b_1$$

$$g_2 (x_1, x_2, . . ., x_n) = b_2$$

‘ ‘ ‘

‘ ‘ ‘

‘ ‘ ‘

$$g_m (x_1, x, {}_2 . . ., x_n) = b_m$$

And K inequality constraints

$$h_1(x_1, x_2, \ldots, x_n) \quad \leq \quad r_1$$

$$h_2(x_1, x_2, \ldots, x_n) \quad \leq \quad r_2$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$h_k(x_1, x_2, \ldots, x_n) \quad \leq \quad r_k$$

In addition, the values of the decision variable $x_1, x_2, \ldots, x_n$ must be integers. No fractional values of $x_1, x_2, \ldots, x_n$ are permitted. In an integer program, the objective function F and the constraint functions $g_1, g_2, \ldots, g_m$ and $h_1, h_2, \ldots, h_k$ may linear or non linear. If we restrict the objective function F and he constraint function $g_1, g_2, \ldots, g_m$ and $h_1, h_2, \ldots, h_k$ to be linear, then we have an integer linear program.

Integer linear programming formalities are much more easily handled in computation than integer (non linear) programming formalities. We seek to an integer programming (IP) formalities of graph colouring, which contains non linear constraint. In this formulation, G is the graph we wish to properly colour. The number of vertices in G is denoted by n and the number of edges in G is denoted by e. We use k to represent the number of colour we wish to use to properly colour G. The values of n and k are known constraints which serve as parameters in the model. The formulation addresses the following two questions;

Given a graph G and a number of colours K, is there a proper K –colouring of G? If such a K – colouring of G exist, what is an example of such coloring.

Integer programming formulation of Graph colouring:

Minimise objective function

Subject to:

1. $1 \le X_i \le k; i = 1,2,\ldots,n$

2. $1 \le \left|X_i - X_j\right|$ for each edge $ij \in E(G)$

3. $X_1, X_2, \ldots, X_n$ are integer – valued.

The above formulation contains n integer variables $x_1, x_2, \ldots, x_n$, each representing one of the n vertices of G. A feasible solution{ x1, x2, . . ., xn} gives a proper K-colouring of G, if indeed one does exist. Three constraints define precisely what it means for a graph to exhibit a proper k-colouring. Constraint 1 maintains that each variable xi must receive a value between I and k. That is to say, each vertex of G must be coloured using a colour from I to K. Constraint 2 further illustrates the definition of a proper colouring of G. For each edge of G, the vertices corresponding to that edge must be coloured using different colours. Finally, constraint 3 requires that the values of the variable be $x_1, x_2, \ldots, x_n$ integers.

Obviously this condition must hold since our k colours are numbered as integers 1, 2, . .., k. If we obtain a solution{ x1, x2, . . ., xn} that satisfies all of the above constraints for a particular graph G given k, then G has a proper k-colouring and $\{x_1, x_2, \ldots, x_n\}$ is such a colouring.

3.6 SEQUENTIAL GRAPH COLOURING ALOGRITHMS

Many algorithms and heuristics exist for colouring approach. Given a graph G = (V, E), the randomly ordered sequential (RND) algorithm randomly orders the vertices so that V = $\{V_1, V_2, \ldots V_n\}$ and then assigns colours to the vertices in the following manner.

The first vertex, V1 is assigned colour number 1. Once the first vertices has been coloured ($1 \leq i \leq n-1$), Vi + 1 is assigned the lowest possible colour number such that no previously coloured vertex adjacent to Vi + 1 has been assigned the same colour number. For any graph G, there exist an ordering of the vertices for which RND will produce an optimal (minimum) colouring of G, while there may exist another particular vertex ordering that lead RND to compute an extremely poor colouring of G. Therefore, the problem of finding an optimal initial ordering of the vertices of a graph is equivalent to the problem of optimally colouring the graph.

This resulted to the development of several new sequential colouring algorithms which differ from RND only in the method of initially ordering the vertices of the graph. Two such algorithms are the Largest-First (LF) and the Smallest-Last (SL). The LF algorithm orders the vertices such that d (Vi) d (Vi + 1) for $1 \leq i \leq n$ where V = $\{V_1, V_2, \ldots, V_n\}$. An SL ordering recursively orders the vertices of smallest degree last. An SL ordering in one in which d $(V_n) = \min_{w\square v} d(w)$, and for n -≥, $i \geq 1$, $d_n(Vi) = \min_{w\square u} d_n(w)$, where U $= V - \{V_n, \ldots, Vi + 1\}$.

Each of these three sequential colouring algorithms requires O (n^3) time an O (n^2) space to colour a graph with n vertices.

Sequential graph colouring algorithms are commonly referred to as greedy algorithms. A greedy graph colouring approach takes each vertex in turn in some particular order and tries to colour the vertex with one of the colours used so far. That is, it tries to add the vertex to one of the existing colours classes. If it is not possible to colour the vertex with any existing colour, then a new colour class is created and the vertex is assigned the colour of that new class.

CHAPTER FOUR

THE SCHEDULE MODEL

4.1 INTRODUCTION

Infant mortality refers to the death of infants and children under the age of five. About twenty-five thousand (25,000) young children die every day mainly from preventable causes. In 2007, 9.2 million children under five died. About half of infant deaths occurred in Africa and Ghana is not left out (UNICEF press release September 12, 2008).

Mother mortality is also another worrying problem facing Hospitals in Ghana. We seek to really schedule maternity ward nurses to shifts in a way that will make every nurse happy and willing to work efficiently to combat or reduce infant and mother mortality rate at Ejura District Hospital, in the Ashanti Region of Ghana.

4.2 OVERVIEW OF THE MODEL

The important task of scheduling nurses at a District Hospital in Ghana is an example of timetabling (shift tabling). Timetabling is the scheduling of a set of related events in a minimal block of time such that no resource is required simultaneously by more than one event. In Hospital scheduling the resources involved, which we assume may be required by no more than one nurse at any particular day, are the Morning Shift, Afternoon Shift and the Night Shift. As mentioned earlier, scheduling (in particular, Hospital scheduling is a practical application of graph colouring.

However, solutions to large-scale practical problems including scheduling problems are often desired and needed much more quickly than exhaustive search algorithm could help to provide. As a result, numerous heuristics have been developed for such problems, which produce near-optimal satisfactory solutions in much less time.

In this work, we shall develop and present a mathematical and computational model with variations for solving Hospital scheduling problems for maternity nurses using techniques and heuristics of graph colouring and a sequentially devised technique in coming out with a near-perfect shift time.

The model involves creating a conflict graph from the assembled data from the maternity ward (Department) of the Ejura District Hospital. Properly colouring the conflict graph and transforming this colouring into a conflict free table of nurses. From these conflicts free nurse table, one can then assign the nurses to different shifts based on the hard constraints of the ward and/or the hospital in general.

At the maternity ward of Ejura District Hospital, it is the norm that nurses are scheduled to different shifts on a particular day every forthnight (two weeks) for the year. When attempting to model and solve nurse scheduling problems (in the ward), it is therefore logical to assume that a particular set of nurses has been designated to be staffed at a particular period and that these nurses are to be assigned appropriate (i.e., non-conflicting) shift slots.

Hospital nurse scheduling problems involves pair wise restrictions on the nurses being scheduled; that is, there exist restriction on which nurses can be scheduled simultaneously and not.

The restrictions involved in creating a shift table of nurses may be divided into two categories as already discussed extensively in chapter one of this thesis. These categories are called essential (hard constraints) and preferential (soft constraints) scheduling conditions.

We recall that preferential scheduling conditions are additional conditions or constraints that need not necessarily be satisfied to produce a legal or legitimate shift table, but if satisfied, may very well produce a more "acceptable" shift table for nurses and/or the ward.

These conditions are requests that should be fulfilled, if possible. However, our study takes into critical consideration of the preferential conditions. That is the preferences of each nurse irrespective of your rank and experience is considered in our study unlike other studies that do not cater for all soft constraints.

The consideration of the preference of nurses results into a conflict graph. The conflict graph when properly coloured (that is application graph colouring), gives a conflict-free , which makes it possible to schedule nurses into conflict-free shift table. Once the conflict-free shift table is obtained, we proceed by applying the hard constraints of the ward and as such a suitable schedule is obtained.

4.3 ASSEMBLING HOSPITAL NURSE DATA

Before the scheduling of a hospital nurse for a particular ward, can take place, we must first assemble a collection of nurses' data for the ward, to serve as input to our problem.

As mention in section 4.2, we assume in our scheduling model that we have a set of nurses that are available to be schedule at a particular point in time and that these nurses are to be assigned appropriate (i.e., non-conflicting) shift slots, subject to a number of essential and preferential scheduling conditions.

Each nurse to be scheduled constitutes a data entry, containing the required or optional information below. Such information is expected to be supplied by either the nurse, the matron of the ward or the hospital administration or some other appropriate sources such as the Ministry of Health (MOH) Ghana.

The required relation is as follows:

- NURSE: The NURSE table contains data of nurses available at the ward for the schedule. It has a field NURSE-ID, which serves as a primary key for the nurse table, YEAR-EXPERIENCE field, contains data of the number of years a nurse has been in services and/or the ward, NURSE-NAME field has the name of a nurse, NURSE-RANK field has the rank of each nurse to be scheduled and NO-OF-NURSES for the total number of nurses available in the ward for the schedule.
- SHIFT: The SHIFT table contains data on different shifts that are available and has fields, MORNING SHIFT which contains nurses scheduled to morning shift, AFTERNOON-SHIFT for the afternoon and NIGHT-SHIFT field for nurses to be assigned to the night shift.

- **DUTYOFF**: The DUTYOFF has data on the off duty each nurse is entitled to. That is a nurse will not go to work. It has fields, DAY-OFF which contains nurses on a day off duty, HOLIDAY-OFF has nurses who do not go to work on public holidays and NIGHT-OFF has nurses who received a number of rest days after a four continuous night shifts.

Below is a diagrammatic representation of all that has been said about the tables.

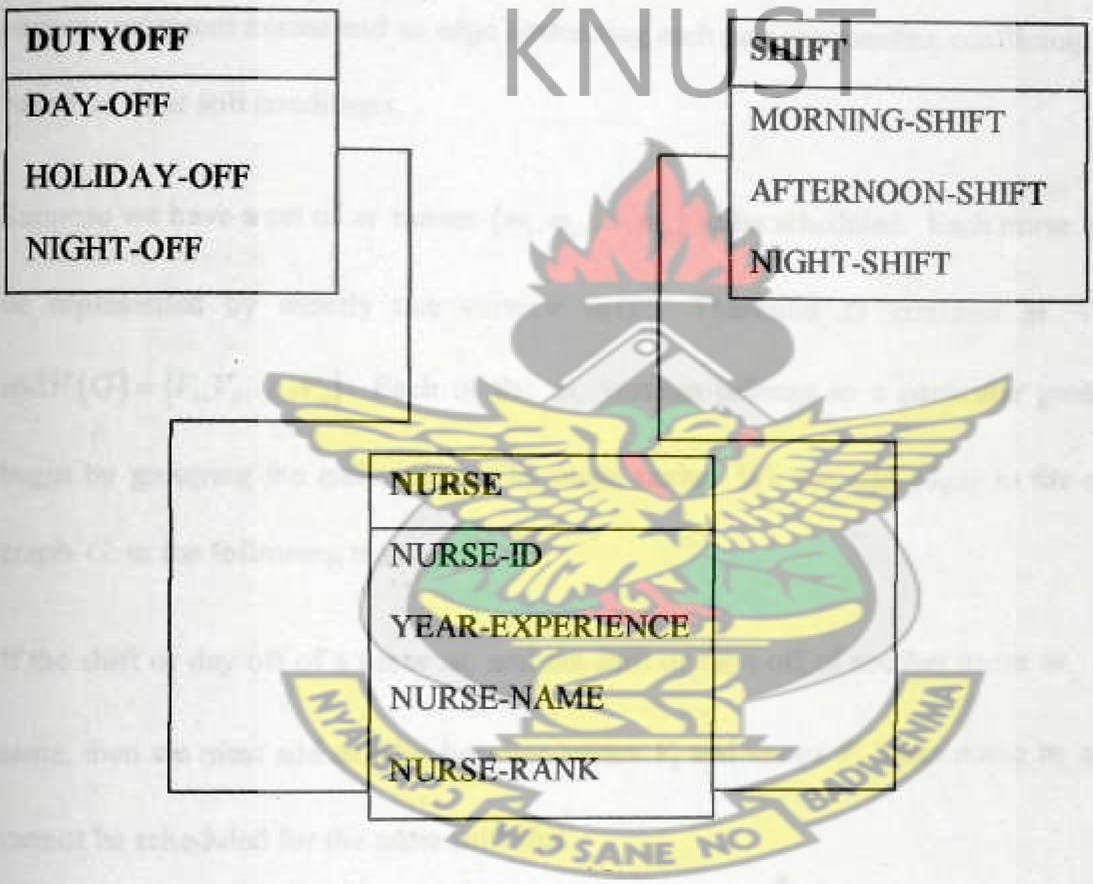


Figure 4.1

This information will be helpful in assigning nurses to available shifts.

4.4 CREATING A NURSE CONFLICT GRAPH

Coming out with a desirable conflict graph is first step is trying to come out with schedule. Once we have gathered all of the necessary nurse data for a given period (fortnight) as described above, we can construct a nurse scheduling conflict graph reflecting the given data.

Recall that in a conflict graph, the vertices represent the items of interest, and in our case vertices represent nurses and an edge connecting each pair representing conflicting nurses based on their soft conditions.

Suppose we have a set of m nurses $\{m_1, m_2, \ldots, m_n\}$ to be scheduled. Each nurse m_i will be represented by exactly one vertex v in G. Therefore G contains m Vertices and $V(G) = \{V_1, V_2, \ldots, V_n\}$. Each of the m_i vertices belongs to a particular group. We begin by grouping the gathered data by nurse ranks. We can add edges to the conflict graph G in the following manner.

If the shift or day off of a nurse m_i and the shift or days off of another nurse m_j are the same, then we must add an edge between vertex V_i and vertex V_j since nurse m_i and m_j cannot be scheduled for the same shift slot.

The diagram below is the conflict graph of nurses in the maternity ward of Ejura District Hospital.

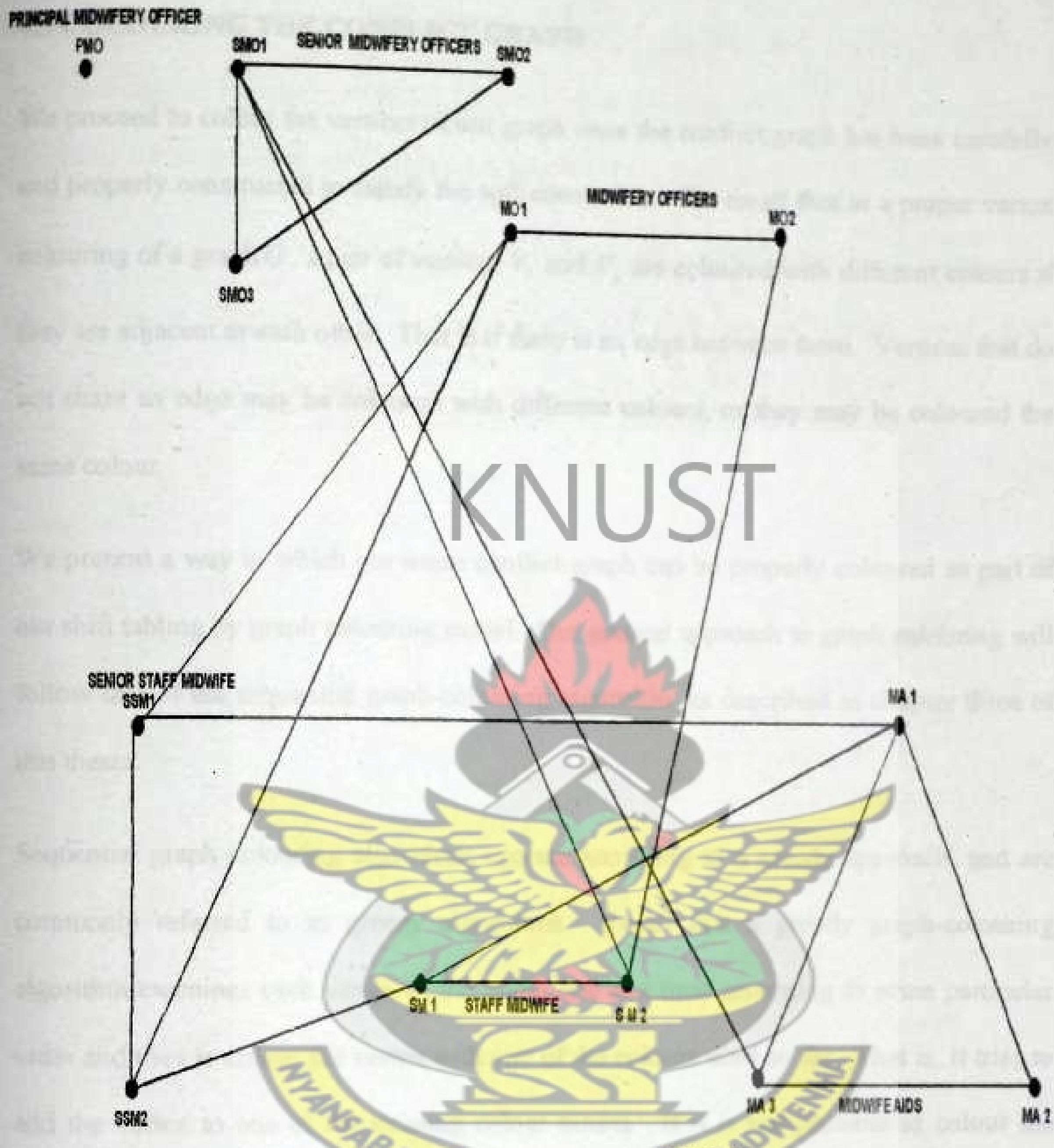

PRINCIPAL MIDWIFERY OFFICER
PMO
SMO1
SENIOR MIDWIFERY OFFICERS
SMO2
SMO3
MIDWIFERY OFFICERS
MO1
MO2
SENIOR STAFF MIDWIFE
SSM1
MA 1
SM 1
STAFF MIDWIFE
SM 2
SSM2
MA 3
MIDWIFE AIDS
MA 2

Figure 4.2

4.5 COLOURING THE CONFLICT GRAPH

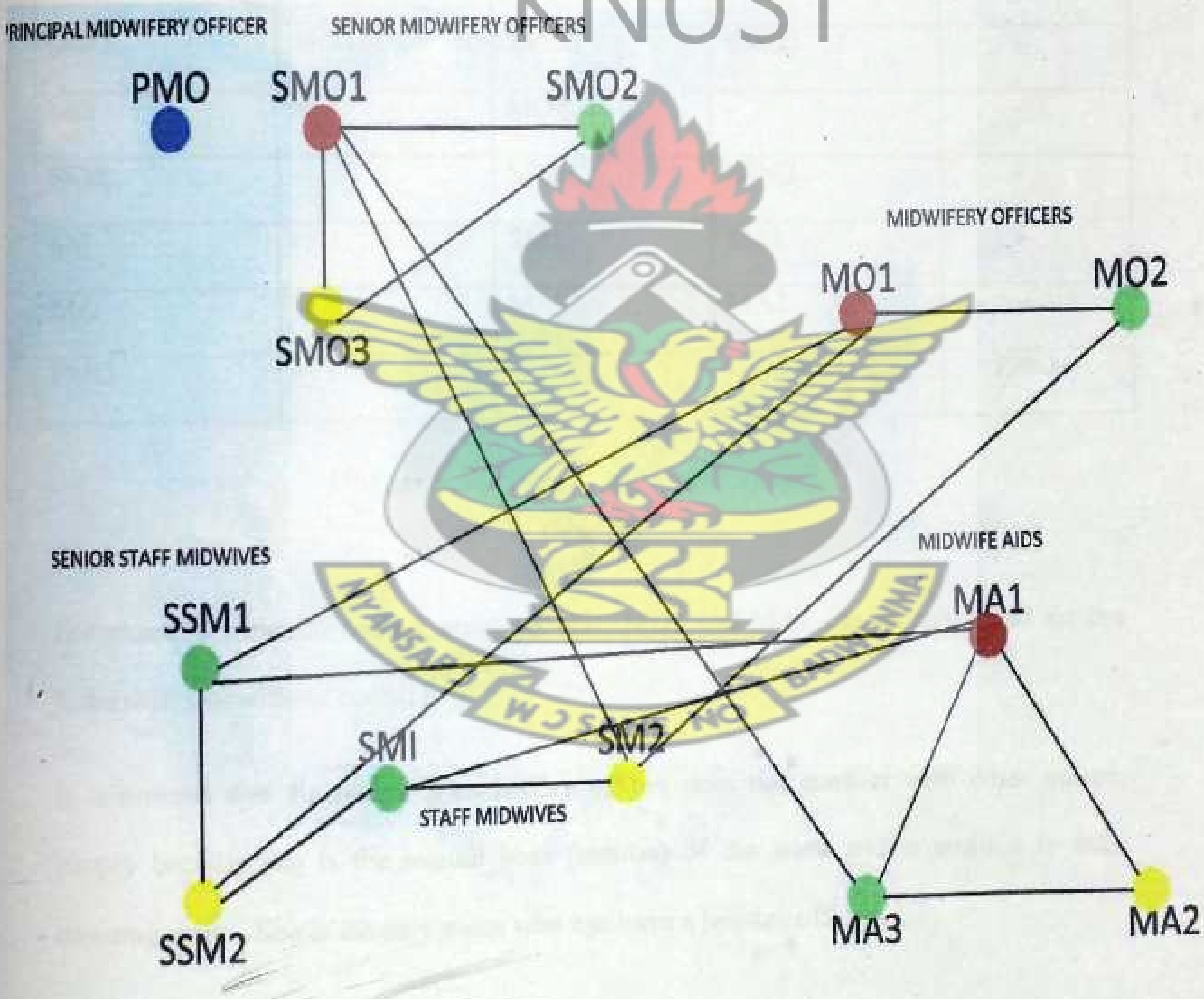
We proceed to colour the vertices of our graph once the conflict graph has been carefully and properly constructed to satisfy the soft constraints. We recall that in a proper vertex colouring of a graph G, a pair of vertices V_i and V_j are coloured with different colours if they are adjacent to each other. That is if there is an edge between them. Vertices that do not share an edge may be coloured with different colours, or they may be coloured the same colour.

We present a way in which our nurse conflict graph can be properly coloured as part of our shift tabling by graph colouring model. Our general approach to graph colouring will follow that of the sequential graph-colouring algorithm, as described in chapter three of this thesis.

Sequential graph colouring algorithms operate according to a greedy approach, and are commonly referred to as greedy algorithms. Recall that a greedy graph-colouring algorithm examines each vertex of the graph one at a time according to some particular order and tries to colour the vertex with one of the colours used so far. That is, it tries to add the vertex to one of the existing colour nurses. If it is not possible to colour the vertex with any existing colour, then a new colour is created and the vertex is assigned the colour. Greedy sequential graph colouring algorithms attempt to properly colour a graph using the maximum number of colours possible.

The Largest-First Search Greedy (LFSG) was implemented. Here, vertices corresponding to nurses who cannot be assigned to the same shift slot due to potential conflicts will be coloured with different colours in our model.

Below is a diagrammatic representation of the coloured conflict graph for nurses at the maternity ward of Ejura District Hospital. Here, alphabets are used to represent colours for the various vertices colouring



4.6 TRANSFORMING THE COLOURING TO A SHIFT TABLE

A technique has been devised to carefully transform the coloured conflict graph into a near-perfect shift table. The table below shows nurses under colours which simply explain that, if the nurses that fall under the same colour are scheduled for the same shift slot, no conflict will arise.

COLOURS NURSES	RED	GREEN	YELLOW	BLUE
SMO	SMO1	SMO2	SMO3	-
MO	MO1	MO2	-	-
SSM	-	SSM1	SSM2	-
SM	-	SM1	SM2	-
MA	MA1	MA3	MA2	-
PMO	-	-	-	PMO

Table 4.1

For example with colour Red, nurses SMO1, MO1 and MA1 can be scheduled for the same shift slot without conflicting.

It is noticed that the principal midwifery officer does not conflict with other nurses, simply because she is the overall boss (matron) of the ward and is entitled to only morning shifts. She is the only nurse who can have a holiday off duty.

We now construct the shift table. A technique for the allocation of nurses to shift table was devised. The allocation follows a format, which is going to be adhered to throughout. Thus "allocation" in this section refers to such a technique.

We denote Morning shift by M, Afternoon shift by A and Night shift by N. Also Day off duty by D/O, Night off duty by N/O and Public holiday off duty by H/O. Before the allocations, we recall the hard constraints like;

(i) Only the Principal Midwifery Officer (PMO) is entitled to Holiday off duty.

(ii) Principal midwifery officer is scheduled for only morning shifts and has day off duties on both Saturdays and Sundays.

(iii) Every nurse is entitled to at least one day off a week.

(iv) Every nurse is entitled to three (3) days off after a night shift.

(v) Every night shift is taken continuous for four conservative days.

(vi) The minimum number of nurses for morning shift should not be less than three. That is $M \geq 3$.

(vii) The number of nurses for both Afternoon and Night should be at least two. That is $A \geq 2$ and $N \geq 2$.

We shall also denote the days of the week i.e. Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday by Mo, Tu, We, Th, Fr, Sa, Su respectively.

We then start constructing our shift table by representing the column with days of the week and rows with the nurses. We assign all nurses under the Red colour on Afternoon shift, Green colour on Morning shift and Yellow colour on Night shift for the start.

COLOURS	WEEK DAYS NURSES	Mo	Tu	We	Th	Fr	Sa	Su
BLUE	PMO	H/O	M	M	M	M	D/O	D/O
RED	SMO1	A	A	A	A	A	A	A
	MO1	A	A	A	A	A	A	A
	MA1	A	A	A	A	A	A	A
GREEN	SMO2	M	M	M	M	M	M	M
	MO2	M	M	M	M	M	M	M
	SSM1	M	M	M	M	M	M	M
	SM1	M	M	M	M	M	M	M
	MA2	M	M	M	M	M	M	M
YELLOW	SMO3	N	N	N	N	N/O	N/O	N/O
	SSM2	N	N	N	N	N/O	N/O	N/O
	SM2	N	N	N	N	N/O	N/O	N/O
	MA3	N	N	N	N	N/O	N/O	N/O

Table 4.2

From Table 4.2, from Friday onwards there must be nurses on night shift since nurses under colour Yellow have finished their night shift and are on Night off. Hence nurses under colour Red takes the task and they receive a day off on Thursday to help them rest and prepare for the night shift.

Nurses under colour Green therefore share the morning and afternoon shifts as seen in Table 4.3

COLOURS	WEEK DAYS NURSES	Mo	Tu	We	Th	Fr	Sa	Su
BLUE	PMO	H/O	M	M	M	M	D/O	D/O
RED	SMO1	A	A	A	D/O	N	N	N
	MO1	A	A	A	D/O	N	N	N
	MA1	A	A	A	D/O	N	N	N
GREEN	SMO2	M	M	M	A	A	M	M
	MO2	M	M	M	M	M	M	M
	SSM1	M	M	M	M	M	M	M
	SM1	M	M	M	A	A	A	A
	MA2	M	M	M	A	A	A	A
YELLOW	SMO3	N	N	N	N	N/O	N/O	N/O
	SSM2	N	N	N	N	N/O	N/O	N/O
	SM2	N	N	N	N	N/O	N/O	N/O
	MA3	N	N	N	N	N/O	N/O	N/O

Table 4.3

We then come to the fact that, a nurse must have at least a day off for a week and $M \geq 3$, $A \geq 2$ and $N \geq 2$ to assign some day off for the nurses. Nurses under Yellow colour have three night days off which are accepted. We assign day(s) off to nurses under colours Red and Green based on the above facts. This is seen in table 4.4.

COLOURS	WEEK DAYS NURSES	Mo	Tu	We	Th	Fr	Sa	Su
BLUE	PMO	H/O	M	M	M	M	D/O	D/O
RED	SMO1	A	A	D/O	D/O	N	N	N
	MO1	D/O	A	A	D/O	N	N	N
	MA1	A	D/O	A	D/O	N	N	N
GREEN	SMO2	D/O	M	M	A	A	M	M
	MO2	M	D/O	D/O	M	M	M	M
	SSM1	D/O	M	D/O	M	M	M	M
	SM1	M	D/O	M	A	A	A	A
	MA2	M	M	D/O	A	A	A	A
YELLOW	SMO3	N	N	N	N	N/O	N/O	N/O
	SSM2	N	N	N	N	N/O	N/O	N/O
	SM2	N	N	N	N	N/O	N/O	N/O
	MA3	N	N	N	N	N/O	N/O	N/O

Table 4.4

From Table 4.4 we can see and deduce the number of nurses for each shift which conform to the hospital hard constraint. Table 4.5 gives a clear picture for the first week.

DAY	MORNING SHIFT $M \geq 3$	AFTERNOON SHIFT $A \geq 2$	NIGHT SHIFT $N \geq 2$
MONDAY	3	2	4
TUESDAY	4	2	4
WEDNESDAY	3	2	4
THURSDAY	3	3	4
FRIDAY	3	3	3
SATURDAY	3	2	3
SUNDAY	3	2	3

Table 4.5

Successfully scheduling nurses for the first week, we proceed to the second week with the same technique and concept. Here nurses under colour Red will complete their night shift on Monday and therefore obtain a three day Night off. Nurses under colour Yellows are schedule for morning shift from Monday to Thursday. Automatically nurses under colour Green share the Afternoon shift and Night shift. But Night shift nurses received a day off before they start their shift. Thus the table becomes.

COLOURS	WEEK DAY NURSES	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr
BLUE	PMO	H/O	M	M	M	M	D/O	D/O	M	M	M	M	M
RED	SMO1	A	A	D/O	D/O	N	N	N	N	N/O	N/O	N/O	
	MO1	D/O	A	A	D/O	N	N	N	N	N/O	N/O	N/O	
	MA1	A	D/O	A	D/O	N	N	N	N	N/O	N/O	N/O	
GREEN	SMO2	D/O	M	M	A	A	M	M	D/O	N	N	N	N
	MO2	M	D/O	D/O	M	M	M	M	A	A	A	A	
	SSM1	D/O	M	D/O	M	M	M	M	A	A	A	A	
	SM1	M	D/O	M	A	A	A	A	D/O	N	N	N	N
	MA2	M	M	D/O	A	A	A	A	D/O	N	N	N	N
YELLOW	SMO3	N	N	N	N	N/O	N/O	N/O	M	M	M	M	
	SSM2	N	N	N	N	N/O	N/O	N/O	M	M	M	M	
	SM2	N	N	N	N	N/O	N/O	N/O	M	M	M	M	
	MA3	N	N	N	N	N/O	N/O	N/O	M	M	M	M	

Table 4.6

Next, Nurses under colour Green who went for night duty receive their night off duty for three days. Nurse's MO2 and SSM1 begin their night shift from Friday to Sunday for the second week, while Afternoon shifts are assigned to nurses under colour Yellow. The shift table at this stage becomes,

OLO RS	WEEK DAYS NURSES	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu
LUE	PMO	H/O	M	M	M	M	D/O	D/O	M	M	M	M	M	D/O	D/O		
ED	SMO1	A	A	D/O	D/O	N	N	N	N	N/O	N/O	N/O	M	M	M		
	MO1	D/O	A	A	D/O	N	N	N	N	N/O	N/O	N/O	M	M	M		
	MA1	A	D/O	A	D/O	N	N	N	N	N/O	N/O	N/O	M	M	M		
REE	SMO2	D/O	M	M	A	A	M	M	D/O	N	N	N	N	N/O	N/O	N/O	
	MO2	M	D/O	D/O	M	M	M	M	A	A	A	A	D/O	N	N	N	N
	SSM1	D/O	M	D/O	M	M	M	M	A	A	A	A	D/O	N	N	N	N
	SM1	M	D/O	M	A	A	A	A	D/O	N	N	N	N	N/O	N/O	N/O	
	MA2	M	M	D/O	A	A	A	A	D/O	N	N	N	N	N/O	N/O	N/O	
ELL W	SMO3	N	N	N	N	N/O	N/O	N/O	M	M	M	M	A	A	A		
	SSM2	N	N	N	N	N/O	N/O	N/O	M	M	M	M	A	A	A		
	SM2	N	N	N	N	N/O	N/O	N/O	M	M	M	M	A	A	A		
	MA3	N	N	N	N	N/O	N/O	N/O	M	M	M	M	A	A	A		

Table 4.7

We then make sure that the hard constraint which state that; every nurse is entitled to at least a day off is satisfied and this is reflected in Table 4.8.

OL OUR	WEEK DAYS NURSES	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu
BLU	PMO	H/O	M	M	M	M	D/O	D/O	M	M	M	M	M	D/O	D/O		
RED	SMO1	A	A	D/O	D/O	N	N	N	N	N/O	N/O	N/O	M	M	M		
	MO1	D/O	A	A	D/O	N	N	N	N	N/O	N/O	N/O	M	M	M		
	MA1	A	D/O	A	D/O	N	N	N	N	N/O	N/O	N/O	M	M	M		
GRE EN	SMO2	D/O	M	M	A	A	M	M	D/O	N	N	N	N	N/O	N/O	N/O	
	MO2	M	D/O	D/O	M	M	M	M	A	A	A	A	D/O	N	N	N	N
	SSM1	D/O	M	D/O	M	M	M	M	A	A	A	A	D/O	N	N	N	N
	SM1	M	D/O	M	A	A	A	A	D/O	N	N	N	N	N/O	N/O	N/O	
	MA2	M	M	D/O	A	A	A	A	D/O	N	N	N	N	N/O	N/O	N/O	
YEL LOW	SMO3	N	N	N	N	N/O	N/O	N/O	D/O	M	M	M	D/O	A	A		
	SSM2	N	N	N	N	N/O	N/O	N/O	M	D/O	M	M	A	D/O	A		
	SM2	N	N	N	N	N/O	N/O	N/O	M	M	D/O	M	A	A	D/O		
	MA3	N	N	N	N	N/O	N/O	N/O	M	M	M	D/O	A	A	D/O		

Table 4.8

CHAPTER FIVE

RESULTS, CONCLUSIONS AND RECOMMENDATIONS

5.1 SHIFT TABLE CONSTRUTED FROM THE PROPOSED METHOD

The first and second weeks shift table for the maternity ward of Ejura District Hospital resulted from the describe method devised in Section 4.6 is in Table 5.1. From Table 4.8, it is clearly seen that the various nurses have been scheduled for the two weeks. We therefore arrange the shift table in terms of rank of the nurses.

WEEK DAYS / NURSES	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su
PMO	H/O	M	M	M	M	D/O	D/O	M	M	M	M	M	D/O	D/O
SMO1	A	A	D/O	D/O	N	N	N	N	N/O	N/O	N/O	M	M	M
SMO2	D/O	M	M	A	A	M	M	D/O	N	N	N	N	N/O	N/O
SMO3	N	N	N	N	N/O	N/O	N/O	D/O	M	M	M	D/O	A	A
MO1	D/O	A	A	D/O	N	N	N	N	N/O	N/O	N/O	M	M	M
MO2	M	D/O	D/O	M	M	M	M	A	A	A	A	D/O	N	N
SSM1	D/O	M	D/O	M	M	M	M	A	A	A	A	D/O	N	N
SSM2	N	N	N	N	N/O	N/O	N/O	M	D/O	M	M	A	D/O	A
SM1	M	D/O	M	A	A	A	A	D/O	N	N	N	N	N/O	N/O
SM2	N	N	N	N	N/O	N/O	N/O	M	M	D/O	M	A	A	D/O
MA1	A	D/O	A	D/O	N	N	N	N	N/O	N/O	N/O	M	M	M
MA2	M	M	D/O	A	A	A	A	D/O	N	N	N	N	N/O	N/O
MA3	N	N	N	N	N/O	N/O	N/O	M	M	M	D/O	A	A	D/O

Table 5.1

5.2 CONCLUSIONS AND RECOMMENDATIONS

The aim of our thesis was to satisfy the various application of graph colouring and coming out with a model in solving the schedule problem for nurses using graph colouring techniques. It clearly has fulfilled most of our essential and preferential conditions present for our shift tabling condition. Essential shifts tabling conditions (also commonly referred to as hard constraints) are conditions or constraints that must be satisfied in order to produce a legal or feasible shift table.

Preferential shift tabling conditions (also commonly referred to as soft constraints) are additional conditions or constraints that need not necessarily be satisfied to produce a legal or legitimate shift table, but if satisfied, may very well produce a more acceptable schedules for nurses and/or ward members.

In this thesis, we discussed the relevance of Hospital scheduling problems as a natural and practical application of graph colouring. The most important contribution of this thesis to the Hospital scheduling problem is the formulation of a new model that helps us eliminate most of the problems encountered in constructing the shift table and then coming out a near-perfect one. The model comprises assembling the course data, creating a course conflict graph based on the assembled data and soft constraints using methods described in section 4.4 of chapter four, performing a proper colouring of the graph in section 4.5 of chapter four transforming the colouring to a conflict-free table and finally assigning nurses to shifts based on the hard constraints.

Once a conflict-free nurse shift table has been constructed, we can use it to create a similarly conflict-free shift for nurses as well.

The schedule officer may very well wish to use our schedule model multiple times, incorporating different sets of preferential conditions each time, until finally arriving at a shift table that is ultimately most suitable to his or her liking.

We can finally conclude that the scheduling-by-graph-colouring model incorporating such specific graph colouring methods and the devised technique will ultimately produce more satisfactory nurse's shifts table for the maternity ward and Ejura District Hospital entirely.

RECOMMENDATIONS

Graph colouring in general is a new study area that researchers and academics can further investigate its applications to real-life pressing problems. The scheduling problem has a large scope that further research has to be carried out to unravel its potential in this area. It would be of great importance to extent this to all wards/departmental levels of Ejura District Hospital and other districts, municipal, regional and teaching hospitals in the country.

The next step would be to fully design program software for the model in this project to help with the easy creation of the shift tables.

REFERENCES

1. Aigner M.(1987) Graphentheoric: Eine Entwickling aus dem 4-Farbenproblem (B. G. Teubner, Stuttgart, 1987)

2. Appel K and Haken W. (1979) Every Planar map is four colourable. Illinois Journal of math, Vol 21:429 – 567, 1979

3. Baki Koyuneu and Mahnut Secir (2004) Student time table by using graph colouring algorithm

4. Bouchand et al., (1986). Ultrametricity transition in the graph colouring problem. Europhys. Lett. 91-98

5. Burke E. K, Elliman D. G. and Weare R. (1995) A university Timetabling system based on Graph Colouring and constraint manipulation.

6. Cheng B. M. W., Lee J. H. M and Wu J. C. K (2003). A constraint-based Nurse Rostering system using a Redundant modeling Approach.

7. Cheng et al., (1997). A nurse rostering system using constraint programming and redundant modeling. Vol. 1, PP 44-54.

8. Duffy et al., (2006). Complexity analysis of a decentralised graph colouring of algorithm. Ireland Mathematics Institute

9. Ellie D'Hondt, (2008). Quantum algorithms for graph colouring problems

10. Fre et al., (2006). An exact method for graph colouring. Computer and operations Research, Vol 33, Issue 8, PP 2189 – 2207

11. Holloway et al., (1993) A generalized Algorithms for Graph-Colouring register allocation .

12. Joseph Culberson and Ian Gent (2001). Frolen development in graph coloring.

13. Juhos et al., (2004). Binary Merge Model Representation of the Graph Colouring Problem.

14. Kenneth H. Rosen (1999) Discrete Mathematics and its applications 4th edition, 1999: 438 – 517.

15. Liu et al., (2002). DNA solution of a graph colouring problem. J. Chem. Info. Comput. Sci 2002, 42(3), PP 524-528

16. Maggie Johnson, graph colouring. CS103B Handout # 20.

17. Marx D. (2004). Graph colouring problems and their applications in scheduling.

18. Murphey et al., (1999) Frequency Assignment Problems. Handbook of combinatorial optimization.

19. Noga Alon, (1993). Restricted colouring of graphs. London math Soc. Lecture Notes series 187.

20. Nolte and Schrader, (2002). An application of simulated annealing to the 3 – colouring problem.

21. Stephen T. Hedetmemi (2002). Fault tolerant distributed coloring algorithms that stabilize in linear time. IEEE IPDPS.

22. Trick M. (1994). Network Resources for colouring a graph.

23. Valls et al., (1996). A graph colouring model for assigning a heterogeneous workforce to a given schedule. European Journal of Operational Research, Elservier B. V.

24. Waters R. J (2005). Graph colouring and Frequency Assignment. Thesis for Degree of Doctor of Philosophy.

25. Wilson R,(2004) Four Colors Suffice: How the map problem was solved (Princeton University Press, 2002)