Thesis submitted for the Degree of Master of Applied Science

DECISION SUPPORT SYSTEMS FOR UNIVERSITY TIMETABLING

by

Eileen Mary SOUTHWELL

Department of

Mathematics, Computing and Operations Research

Footscray Institute of Technology

VICTORIA UNIVERSITY OF TECHNOLOGY

1991

Southwell, Mary
Decision support systems
for university timetabling

This thesis contains no material that has been accepted for the award of any other degree or diploma in any University or Tertiary Institute. To the best of my knowledge and belief it contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

[redacted]

(E. MARY SOUTHWELL)

July 1991

ABSTRACT

Decision Support Systems developed for examination and class timetabling at a University are described, and the organisational and human factors behind computerised timetabling are discussed. An original and practical method using Graph Theory produces from current enrolments, examination schedules which are clash-free to students and acceptable to staff. Simple data base methods are provided for class and room scheduling and recording. The system has been in use at Footscray Institute of Technology for several years.

Descriptors

Decision Support Systems

Examination Timetabling - University

Class Timetabling - University

Room Scheduling - University.

ACKNOWLEDGMENTS

I am indebted to Dr.P.Cerone, Dr.R.E.Johnston and Mr.I.W.Murray of Footscray Institute of Technology for their advice and assistance.

The major part of the project has been undertaken within the Institute, and I thank Mr.M.J.Halls, Registrar, and Mr.B.Wise, Dean of the Faculty of Business, for permission to use enrolment data.

For the part of the project undertaken overseas, I am grateful to the people at several universities who gave their time for interviews and for their hospitality.

My thanks also go to Mr R.Anderson, Mr.I.H.Hilton, and Mr.W.Alwast for their assistance, and to my husband Gordon and daughter Bridget for their patience and encouragement.

CONTENTS.

DEFINITIONS

Class Timetable. A schedule keyed on subject and recording the group, day, time, room, and (optionally) lecturer.

Course. The degree for which a student is enrolled, e.g. Bachelor of Business (Accounting). Known as a Major in U.S.A. and Canada.

Department. The basic organisational unit at F.I.T., e.g. the Department of Mathematics, Computing and Operations Research.

Enrolment. The act of a student in enrolling for a course or subject for the following semester (registration in U.S.A./Canada).

Examination Timetable. A schedule for the examination period, relating the subject, day, time and room.

Faculty. There are four Faculties at F.I.T., including Applied Science and Business. See also Lecturer.

F.I.T. or FIT. Footscray Institute of Technology.

Institute. Footscray Institute of Technology, which with the Western Institute makes up the Victoria University of Technology.

Lecture. The students are addressed by a lecturer, typically in a large room. A stream is a partition of a lecture.

Lecturer. A teacher facing a class, whether he/she be professor, lecturer or tutor. The lecturers together make up the Faculty in U.S.A./Canada terminology.

Room Schedule. A schedule keyed on room, and recording day, time and subject.

Semester. A period of 14 weeks of classes, followed by examinations. At F.I.T. there are two regular semesters in a year, plus a shorter Summer semester.

Subject. The subjects which make up the course e.g. Accounting A, Accounting B. Known as Course in U.S.A./Canada.

Teaching Hours. At F.I.T., Monday to Friday, 8.00 a.m. to 9.00 p.m.

Timetable. Synonymous with schedule. The Round 1 timetable precedes student enrolment, and the Round 2 timetable is published shortly before the start of the semester.

Tutorial/Seminar/Laboratory. The students take part in interactive teaching, in a group of 16 to 24.

University. A University, Institute of Technology, College of Advanced Education, or other Tertiary Institution.

1. INTRODUCTION.

1.1. The Scope of this Work.

This thesis contains a review of the literature on timetabling, a description of the terminology of Graph Theory and Decision Support systems, the original work of a Decision Support System for Timetabling implemented at Footscray Institute of Technology, and a review of some practical systems in operation at other Tertiary Institutions.

1.2. The Nature of Timetabling.

The purpose of timetabling is to bring staff and students together in an appropriate room at an appropriate time to hold regular classes and formal examinations. Processes which are considered in connection with timetabling are the forecasting of student, course and subject numbers, the registration or enrolment of students, and the production of schedules for rooms and of timetables for staff and students.

In a tertiary institution a "subject-scheduling" model is used for both class and examination timetabling, (the subject is scheduled against time and room). In a primary or secondary school a "class-teacher" model is used, (the class is scheduled against teacher and room). Whichever model is used, the same three distinct phases in the timetabling process appear, the definition of curricula for each class, the assignment of resources to each class, and the development of a timetable.

Traditionally, timetabling has been done by individual teachers stating their requirements for times and rooms. This method ceases to be satisfactory when there is a rapid increase in student numbers or other great pressure on resources. Positive steps must then be taken to identify and to improve the allocation of resources.

1.3. A Decision Support System for Resource Allocation.

Generally, the first step in developing a DSS is to survey and record the extent of the resources and of the requirements for their use, and to review the existing methods and policies of allocation. The second step is to research the technological advances available, and to select a methodology of decision making that will be acceptable in the organisation. The third step is to design and implement the chosen system, and to maintain it over a number of years. Only then may the system be said to be successful.

1.4. Review of the Literature on Timetabling.

Many articles on timetabling and related matters appeared in the years 1960 to 1980, sparked off by the belief that the newly introduced computer would transform timetabling from an art form to a scientific pursuit, though, at the time, the computer could not handle large practical problems. A lesser number of articles appeared in the 1980's, but with a more practical approach. The articles from 1960 to 1990 are reviewed in Chapter 2, and provide many interesting ideas for timetabling.

1.5. The Theoretical Background to Decision Support Systems.

A description of the theory of Decision Support Systems and of other related systems such as Management Information Systems and Expert Systems is given in Chapter 3. The prescription is to give support to the decision-maker, rather than to solve the problem straight out by some mathematical means.

The development of fast micro-computers has given the manager the ability to determine multiple alternative solutions to real problems, using quantitative techniques, decision making tools and qualitative reasoning. The ability can be applied to the problems of timetabling in tertiary institutions.

1.6. Graph Theory and Terminology.

This thesis includes significant sections on the practical development of a working DSS. In the development of the system for examination timetabling (reported in Chapter 11), considerable use is made of Graph Theory. The necessary general description of the terminology is contained in Chapter 4, followed by a more detailed analysis of the methods of solution of the graph coloring, travelling salesman and clique problems in Chapters 5, 6 and 7.

1.7. DSS for Timetabling, a Case Study at F.I.T.

In 1982, each department at FIT produced its own class and examination timetables for each semester. The construction of timetables took much work, and yet it was haphazard and often late, there was difficulty in finding enough classrooms at popular times, and reports based on the records were unreliable.

Five main areas of work were defined :-

1. The construction of class timetables and room schedules by departments.

2. The assembly, recording and reporting of room schedules by the Institute.

3. The construction of examination timetables by departments.

4. The assembly, recording and reporting of examination timetables by the Institute.

5. The enrolment of students, in a more detailed system supplementary to that operated by the Institute.

A system covering these five areas of work was developed as part of the work requirements for this degree. The overall system is described in Chapter 8. Individual, but correlated, systems which facilitate recording, reporting and decision making in each area are contained in Chapters 9 to 13. Following the development, the systems have subsequently been installed on micro-computers in the Faculty of Business and in the Institute Administration and complement the Student Record System on the main-frame computer.

1.8. DSS for Timetabling - A Review of Other Systems.

In parallel with the development of a system for FIT, some visits were made to sites where DSS or computerised timetabling systems had been implemented. These visits are recorded in Chapter 14, and some general comments are made here :-

1. The work may be instigated in direct response to an administrative need, or as an academic research project.
2. The research part of the work takes a long time, and the system may fail, or not even be implemented, when the instigator moves on.
3. The system may have to be rewritten before, during, or after implementation, due to technological change.
4. The system can only be implemented if the management provides the necessary resources.
5. Entrenched power structures and attitudes about academic freedom may have to be overcome.
6. A successful system may be small, or large and modular, but not large and monolithic. A system covering too small an area may be affected by technical changes in other areas and be ineffectual, a system covering too large an area may come under several power brokers, and thus fail.
7. All of the successful systems have been constructed in isolation, though the authors may have become aware of other work during the later part of the construction or during the implementation phase.

These comments apply generally to the systems in this thesis, as well as the systems seen at other sites.

1.9. Discussion and Conclusion.

Finally, in Chapter 15, there is a short discussion on the matters raised in the earlier part of the work, and some possible topics for future work are outlined.

2. REVIEW OF THE LITERATURE ON TIMETABLING.

2.1 Introduction.

Academics and administrators from a wide variety of disciplines, including economics, computing, management, mathematics, operations research, industrial engineering and education, have contributed to the literature on timetabling. Reports, review articles and bibliographies, case studies, and theoretical approaches to the solution of combinatorial optimisation problems are found.

The literature mentioned in this chapter covers a period from 1964 to 1990, and has been selected from a long list of publications as being an important development in some aspect of tertiary timetabling. Attention is drawn to the particular features of each paper.

2.2. Bibliographies.

1. Bibliographies provide a detailed list of publications on the topic, together with comments on some particular aspects of timetabling.

2. The work of Schmidt and Ströhlein (1980), provided a comprehensive review of the literature on timetabling and related topics to that date. Comments were made on the nature of the timetabling problem, modelling, operations research approaches, computer applications and future trends.

3. Two distinct phases in the timetabling process were identified by deWerra (1985); first the definition of curricula for each class and assignment of resources to classes, and second, the development of a timetable compatible with the requirements in the first phase. Most of the literature reviewed by deWerra dealt with the second phase, either as the class-teacher model (for school class timetables), or as the course-scheduling model (for tertiary class timetables or for examination timetables).

4. Together with a review on examination scheduling literature, some advice on the selection of algorithms was given by Carter (1986). The relative importance of various constraints (e.g. completely conflict-free, minimised violation of secondary constraints, or presence of pre-assignments), affect the choice of the scheduling algorithm.

5. In a survey of timetabling in Germany by Junginger (1986), a mathematical formulation of the school timetabling problem, a heuristic algorithm for constructing timetables, and some experience in the implementation of actual systems in schools were described.

2.3. Early Papers.

The early, and indeed many of the later papers were closely related to timetabling by hand, or showed an incremental approach to the improvement of the previous year's timetable. Much ingenuity was used to exploit the structure of the data to obtain reasonable solutions.

Early work which is of significance in the development of the work of this thesis is that of Broder (1964), Cole (1964), Barraclough (1965), Almond (1965, 1969), Lions (1966), Welsh and Powell (1967), Wood (1968, 1969), Lawrie (1969), Marsten (1974), Dyer and Mulvey (1976), Devine and Kumin (1976), Carter (1978), Desroches and Laporte (1978) and White (1979).

2.4. More Recent Papers.

1. The PhD thesis of Hemmerling (1974) is a lengthy and detailed treatment of the mathematical set-theory background to a difficult practical problem, namely the generation by computer of timetables for South Australian secondary schools.

2. A number of papers by Carter are referenced in various chapters of this thesis. Carter (1979) describes a situation in final examination scheduling for correspondence students, where only two sessions were available but 4000 students were each sitting two courses out of a possible 200 courses.

Carter (1983) used zero-one quadratic programming in a decomposition algorithm for the timetabling problem, and applied it to the programming of multi-section courses. Carter (1987) provided, as a Lagrangian Relaxation approach, another practical algorithm to use for classroom assignment in a room scheduling and student registration system.

3. The work of a number of authors will be mentioned briefly :-
By the application of a graph coloring method, Mehta (1981) produced a conflict free examination timetable in a college of 750 students. Romero (1982), introducing a computer assisted

participative procedure for examination scheduling, recorded the results and reported conflicts, but made no attempt at optimisation. White and Haddad (1983) studied the case where certain sub-sets of examinations must be scheduled into certain timeslots (e.g. where there are both day and evening courses), and used a heuristic method to minimise the overall number of consecutive examinations, and so optimise the schedule.

Tripathy (1984), using binary integer programming, was able to assign 400 subjects to 30 periods of the week for school timetabling. McClure and Wells (1984) described an integer programming model for the assignment of complete teaching schedules to academic staff. Chahal and deWerra (1989) proposing an interactive system for constructing timetables on a micro-computer, based the system on a simple network, determined a collection of path packings, and found a feasible solution for a 13-teacher problem.

4. Laporte and Desroches (1982) described a computerised procedure for constructing examination timetables in universities. There were no conflicts, and the examinations were spread out as evenly as possible for most students. Later (1984) they used a polynomially bounded and very efficient backtracking algorithm on data from five programs with some overlapping courses. In other work (1986) they solved the problem of allocating students to course sections in an Engineering school of 2800 students. Timetables were published, courses were selected by the students, and the sections were allocated by

computer. Clashes were minimised and the course sections were balanced in size.

5. Several papers by Lotfi and others have built up over the years to a full scale implemented system for examination timetabling with room constraints. A Graph Coloring algorithm for large scale scheduling problems due to Lotfi and Sarin (1986) could be applied to examination scheduling, and was said to produce good sub-optimal solutions with very little computational effort. The second phase of the examination problem was solved using a Lagrangian Relaxation approach by Arani, Karwan and Lotfi (1988), the third phase was solved by Arani and Lotfi (1989), and the package was implemented by Lotfi and Cerveny (1989).

6. In a Decision Support System, Glassey and Mizrach (1986) used an integer program both for rooming classes and for advising of alternative times for classes that could not be accommodated. This package was also implemented.

7. Ferland, Babin and Aubin (1986) described an interactive system for the development of course schedules, taking into account room limitations, course sections and student pre-assignments. Later, Aubin and Ferland (1989) included the procedures to handle timetabling and student grouping as an assignment problem.

8. In a case study on the impact of automated timetabling on universities, Sabin and Winter (1986) dealt in detail with a fully computerised registration system for first-year students. The influence of the University management structure on the success of the system was stressed.

3. DECISION SUPPORT SYSTEM THEORY.

3.1. Introduction.

From the time of the introduction of the computer to commercial organisations, the prime use was to replace manual data processing. Transaction processing operations were run in batches, leading to the update of one or more files. At the same time, transactions could be counted, totals updated, and reports prepared on the progress of the batch run and on the state of the files.

When the reports became standardised as to content, and, due to improved printer technology, could be generated at ever faster speeds, the idea appeared of an information system to provide comprehensive information for managers. The idea was extended to provide an interactive system which could combine quantitative analysis with qualitative tools for decision making.

It is now possible, by using quantitative techniques together with decision making tools and qualitative reasoning, to determine multiple alternative solutions to real problems such as timetabling and scheduling problems in tertiary institutions.

3.2. Management Information Systems (MIS).

A suggested procedure for designing a MIS, due to Ackoff (1967) is to identify and analyse each important managerial decision required by the organisation, and to flowchart the relationship between the decisions.

Three types of managerial decision are identified :-

1. Those for which adequate models with optimal solutions are available. The decision process could be incorporated into the information system.

2. Those for which adequate models, but not optimal solutions, are available. Heuristic solution procedures could be provided, or a simulation performed. (Note, most of the problems in timetabling are of this type.)

3. Those for which adequate models cannot be constructed because it is not known what data is relevant, but which could be converted to type 2 by a guess at the controlling variables and a test of the model.

Ackoff advised that, to improve performance, decisions with overlapping informational requirements should be grouped together under a single manager, and that the MIS should be designed to be flexible and adaptive, with frequent review and revision. An explanatory model of the decision process should be constructed and tested to determine the information required. The decision rules and performance feedback should be included in the decision process for complex problems. Enough detail should be included to enable the decision maker both to understand the system and to not misunderstand the implications of potential changes.

3.3. Decision Support Systems (DSS).

Keen (1980) suggested that starting at an understanding of decision making, and working backwards to technology by way of MIS and Management Science, a manager can draw on a range of technological building blocks to develop a valuable support system.

The rules for building a DSS are as follows :-

1. Decide what is the decision or task.
2. Review how the manager carries it out.
3. Review the data that is used.
4. Consider how a DSS can make the process more effective.
5. Plan in a modular fashion, but be flexible.
6. Incorporate the manager's existing analytical methods.
7. Make the user-interface intelligible for input and command.
8. Provide intelligible output-screens and relevant printed reports.

Many successful DSS were developed and implemented in the 1980's, with the manager able to apply qualitative tradeoffs and subjective assessments to a structured or quantitative decision. It was noted by Little (1979) that some of the DSS would later be abandoned after staff or organisational changes, and some would never be implemented because of the internal ramifications of the reduction of free movement to management or the loss of power to the central computing management. According to Highsmith (1987), other DSS would fail because new technology (hardware or software) crucial to the plan was never introduced.

3.4. Management Science (MS).

Keen and Morton (1978) stated that a single solution (optimal in some respect) could be derived by traditional parameters, constraints and relationships identified, a single solution, optimal in some respect, could be derived. The impact was mostly on generating better solutions for certain specified structured problems.

However, the OR / MS approach does not provide, in many cases, an adequate methodology for the analysis of management decisions, and new approaches were tried , as identified by Watson and Buede (1987) in their rules for DSS (additional to those in para.3.3) :-

1. A set of rules for decision making are defined to state what it is to be rational in the face of uncertainty.

2. The rules address the values of the problem owners, that is, their preferences and perceptions.

3.5. Executive Information Systems (EIS).

Executive Information Systems, as described by Gray and others (1989), are designed to give top management on-line access to information about the firm's current activities. They differ from MIS and DSS in that they are used directly by top management without the assistance of intermediaries. They are designed with management's critical success factors in mind, and they use state-of-the-art graphics, communications, and data storage and retrieval methods. They provide few analysis facilities, and are not valuable for decision-making.

3.6. Expert Support Systems (ESS).

Expert systems, a branch of Artificial Intelligence, appear to be a logical extension of DSS, particularly in the area of the solution of difficult problems.

Gray et al. (1989) described expert systems as "techniques which can be used to preserve and disseminate scarce expertise by encoding the relevant knowledge of the expert and making this experience available as a resource to the less experienced person". The systems can also be used to solve problems that thwart traditional programming techniques. ESS pair the expert system and the human in such a way that the expert system provides the knowledge and the reasoning steps, and the human provides overall problem solving direction.

The development of expert systems was based on a descriptive theory of human problem solving, and so the system is a representation of expertise, that is of the knowledge acquired by humans through practice and learning. Expert systems are tailor-made for specific and narrowly defined problem domains in which the inter-relationship between the variables is complex. Klein and Methlie (1990) indicated that the system should draw conclusions and explain its reasoning, and thus replicate to inexpert users the skill of the expert.

3.7. Optimisation and Rationality.

In a formal model of decision making, economic man behaves rationally towards the goal of perfect utility maximisation by calculating the consequences for each alternative decision, ranking the consequences according to preference, and finally computing the optimal decision. In practice a concept of bounded rationality applies, where costs of intelligence and of information processing, and imperfections in perceiving information are taken into account.

An alternative to the rational theory is the policy of "muddling-through", a behaviour by which the decision maker moves incrementally, searching for alternatives which are only slightly different from existing solutions, as explained by Lindblom (1979). Even for the rational decision-maker, complications exist where the goals are vague, problematic, inconsistent or unstable, or where there are ill-defined or late-appearing constraints on the solution, so "muddling-through" may be the best procedure.

In an optimising method, a "best" solution is sought by converging to a maximum (or minimum) value of a scalar function taken to represent the decision maker's goal. In the alternative Better Than Most, or BTM, method, due to Baum and Carlson (1979), solutions are randomly generated, examined for feasibility, and ranked according to a mathematical representation of the decision maker's goal. By generating 100 solutions there is 99.4% confidence of getting at least one solution in the top 5%.

Another reference to a "good" solution, or "admissible" solution was by Hernandez and Proth (1982), who suggested that it was often possible to find a "good" solution to production problems when their peculiarities were taken into account.

However, if the "good" solution is a local one, the manager should check that its effect is not globally "bad".

A further warning, which particularly applies to heuristic solution procedures, from Williams (1974), is that if procedures work, they should be left alone, and not too much time spent on possibly fruitless efforts to improve performance.

3.8. Heuristic Algorithms.

A heuristic method is defined as a procedure for solving problems by an intuitive approach in which the structure of the problem is interpreted and exploited intelligently to obtain a reasonable solution.

The reasons for using a heuristic include :-

1. The mathematical problem is such that an analytical or iterative solution procedure is unknown, or, if known, is computationally prohibitive to use.
2. The heuristic may be simpler to understand than an optimal solution procedure.
3. The heuristic may be used to obtain an initial feasible solution or a good starting bound for the optimal solution procedure.

A good quality heuristic, according to Silver et al. (1980) in their comprehensive "Tutorial on Heuristic Methods", has the following properties, and raises the following questions :-

1. Has a realistic computational effort.
2. Gives a solution that is close to the optimum on average. (Q. How is the optimum determined ?).

3. Has a low chance of a very poor solution. (Q. What are the conditions leading to a poor solution ?).

4. Is as simple as possible.

According to Fisher (1980) the performance of a heuristic can be predicted by empirical testing, worst-case analysis or probabalistic analysis. The worst-case study examines the maximum deviation from optimality that could occur when a specified heuristic is examined within a given problem class.

Most worst-case bounds are rather large when compared with observed average performance, or with performance which most users would regard as acceptable. The best worst-case performance for the Travelling Salesman problem is a heuristic of Christofides, tightened somewhat further by Cornuéjols and Nemhauser (1978), but even this gives a ratio of 3/2 for the bound to observed performance.

The aim, then, is to match the problem and the heuristic to obtain better performance than the worst-case, in other words to interpret and exploit the structure of a problem.

3.9. Interactive Scheduling.

It was hypothesised that a symbiotic relationship between the human being and the electronic computer would be more powerful at certain types of problem solving activities than either would be alone. In the scheduling situation, however, a gap between the promise and the practice of interactive systems was noted by Godin (1978).

The Management Decision System - MDS, proposed by Morton (1971), included the use of a powerful CRT interactive system to manipulate a large data base to assist managers in unstructured decision making. The "assistance" was later to turn to "support", as in Carlson (1977), "recent improvements in low-cost remote computing,minicomputers, terminals and data management software have allowed us to provide a degree of support hardly conceivable as a widespread reality a few years ago."

Several of the papers mentioned in Chapter 2 referred to interactive scheduling, and recent developments in micro computers have allowed the development of commercial systems, such as the "Time Chart" software for building and printing timetables by Alen Mawson of Amig Systems (1988). A complete timetable for a school or college may be set up, stored, altered, and printed out, interactively. As the data is entered, the program indicates clashes for rooms, teachers or students, thus helping with the design as well as the presentation of the timetable. Some effort is required to learn to use this system.

3.10. Characteristics of a Desirable DSS.

In conclusion, several desirable characteristics may be drawn from the types of systems already described.

1. Decisions with overlapping informational requirements should be grouped together under a single manager, and the manager should be given the responsibility, the power and the means to carry out the task.

2. The system should be modular and be flexible to changing requirements and to changes imposed from other systems.

3. Communication between man and machine should be interactive, but kept to the minimum required for the running of the program.

4. Data should, wherever possible, come from existing data bases, a minimum of parameters should be entered during the run (to reduce transcription and keying errors).

5. The system should not be over complicated but, at the same time, may require very precise and sophisticated techniques in certain areas.

6. Programs should be designed for specific purposes and conditions if they are to be of value for decision-making.

7. Because of the overlapping and interlocking requirements of various parts of the system, a satisfactory solution may be more important than an optimal one, and a feasible solution satisfactory to the users (or even, not unsatisfactory to the users), may be the most desirable and satisfying outcome.

4. GRAPH THEORY AND TERMINOLOGY.

4.1. Introduction.

The combinatorial and graph theory required for the development and solution of the examination timetable problem is described in this part of the thesis.

Firstly, the terminology of GRAPH THEORY is described, (section 4.2). The general properties of a graph are set out because the data for the examination problem may be described as a graph where the set of vertices is the set of subjects being studied by all students, and the set of links joins any pair of subjects being studied concurrently by any student.

Secondly, GRAPH COLORING is described in some detail, (chapter 5), the application being to divide the vertices into sets in which no two vertices are linked. The early summary is due mainly to Christofides (1975), and the later descriptions of the structure and performance of the algorithmic or heuristic methods of solution to Brélaz (1979), Carter (1986), and other writers.

Once the vertices have been divided into sets, it is necessary to arrange the sets in an appropriate order by a TRAVELLING SALESMAN algorithm. The heuristic and bounding procedures required for a solution to this problem have been well described by Boyd, Pulleyblank and Cornuéjols (1988), and are contained in chapter 6.

Finally, some consideration is given to CLIQUE THEORY, the determination of the largest number of vertices which are totally connected, and so must lie in different sets in the graph coloring problem (chapter 7). The theory is important in a situation where the number of sets is constrained, or where a solution to a practical problem has exceeded the number of sets allowed.

4.2. Graph Theory and Terminology.

1. The terminology used for Graph Theory is, unless otherwise indicated, that due to Christofides (1975).

2. A graph G is fully defined by the doublet (X,A) where

$\{x_1, x_2, \ldots\ldots x_n\}$ is the set X of points, and

$\{a_1, a_2, \ldots\ldots a_m\}$ is the set A of lines joining some or all of the points.

Lines with direction are arcs, and the resultant graph is a directed graph . Lines without direction are links and a graph formed entirely of links is a non-directed graph.

The non-directed counterpart to the graph G = (X,A) when the direction of the arcs in A is disregarded is

$$\bar{G} = (X, \bar{A}).$$

The terms edge for an arc or link, and vertex or node for a point are sometimes used by writers other than Christofides, and are equivalent for the purpose of this thesis.

3. An alternative description of the graph G is by the doublet (X,T), where the correspondence T describes how the vertices are related to each other, e.g.

$T(x_1) = \{x_2, x_5\}$ indicates that the final vertices of arcs whose initial vertex is x_1 are x_2 and x_5.

The inverse correspondence $T^{-1}(x_i)$ is the set of the initial vertices of arcs whose final vertex is x_i.

In a non-directed graph the correspondence T is that of an equivalent directed graph in which every link is replaced by two arcs in opposite directions, and, in this case,

$T^{-1}(x_i) = T(x_i)$ for all $x_i \in X$.

4. A path, in a directed graph, is a sequence of arcs in which the final vertex of each arc is the initial vertex of the next arc in the sequence. Two arcs with a common terminal vertex are termed adjacent. Vertices x_i and x_j are adjacent if either arc (x_i, x_j) or arc (x_j, x_i), or both, exist. A simple path does not use the same arc more than once; an elementary path does not use the same vertex more than once (and, by definition, must be simple).

5. A chain is the non-directed counterpart of a path, and may similarly be simple or elementary. A chain may be represented by a sequence of links,

$\overline{a_2}$, $\overline{a_4}$, $\overline{a_6}$ **where $\overline{a}$ is a non-directed link, or, more**

usefully, by a sequence of vertices, $\{x_2, x_5, x_8, x_6\}$.

6. Weights c_{ij} may be associated with arc (x_i, x_j) in which case the graph is said to be arc-weighted, or weights v_i may be associated with vertex x_i, when the graph is vertex-weighted.

For a path $\mu(a_1, a_2, \ldots a_q)$, the length or cost of the path $\ell(\mu) = \sum_{(x_i,x_j) \text{ in } \mu} c_{ij}$ is the sum of the arc-weights.

The cardinality of the path μ is q, the number of arcs in the path.

7. A loop is an arc where the initial and final vertices are the same, ie (x_i, x_i).

A circuit is a path where the initial vertex a_1 and the final vertex a_q are the same. A circuit may be simple or elementary.

A cycle is the non-directed counterpart of a circuit, i.e. a chain with the initial and final vertices the same, and may likewise be simple or elementary.

8. The outdegree of a vertex, x_i, is the number of arcs with x_i as the initial vertex. The indegree of a vertex is the number of arcs with x_i as the final vertex. They are referred to as $d_o(x_i)$ and $d_t(x_i)$ respectively where,

$d_o(x_i) = |T(x_i)|$ and $d_t(x_i) = |T^{-1}(x_i)|$.

The sum of the outdegrees = the sum of the indegrees of all vertices = the total number of arcs of the graph G.

$$\sum_1^n d_0(x_i) = \sum_1^n d_t(x_i) = m$$

where n = number of vertices and m = number of arcs.

For a non-directed graph G = (X,T) the degree of vertex x_i is $d(x_i) = |T(x_i)|$, written as d_i.

9. A partial graph of $G=(X,A)$ is (X,A_p) where $A_p \subset A$, that is the same number of vertices but a lesser number (subset) of arcs.

A subgraph (X_a,T) with $X_a \subset X$ has a subset of the original set of vertices, but contains all the original arcs whose initial and final vertices are both within the subset. A partial subgraph is reduced by vertices, then by arcs.

10. A graph $G = (X,A)$ is complete if for every pair of vertices there is a link, that is there is at least one arc joining every pair of vertices. A complete non-directed graph on n vertices is known as K_n , and contains $n(n-1)/2$ links.

A graph is symmetric if the set A contains the opposite of every arc, e.g. both (x_i,x_j) and (x_j,x_i) are included.

A graph is antisymmetric if any opposite arc is not included, if $(x_i,x_j) \in A$, then $(x_j,x_i) \in A$. An antisymmetric graph cannot contain any loops. A complete antisymmetric graph is called a tournament.

11. A non-directed graph, G is bipartite if the set X of vertices can be partitioned into two subsets X_a and X_b so that all arcs have one terminal vertex in X_a and the other vertex in X_b. that is, $G = (X_a \cup X_b , A)$.

A complete bipartite graph with n vertices has n/2 vertices in each subset each joined to all the vertices in the other subset but disconnected from any vertex in the same subset. An example is $K_{3,3}$.

12. A planar graph can be drawn on a plane or sphere so that no two arcs intersect each other except at vertices. The complete graph K_5 and the complete bipartite graph $K_{3,3}$ are non-planar and

are known as Kuratowski Graphs. The properties of planar graphs are treated in detail by Nishizeki and Chiba (1988).

13. The <u>Adjacency Matrix</u>, A is a convenient representation of T as follows :-

$A = [a_{ij}]$, an n x n matrix, where n is the number of vertices; $a_{ij} = 1$ if an arc exists from x_i to x_j, else 0.

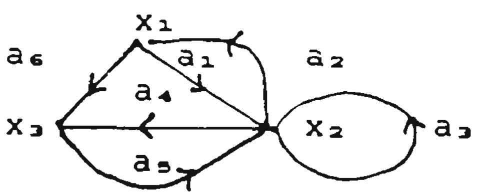
A 1 on the diagonal of the matrix means a loop.

The horizontal sum of the rows of the matrix is the outdegree, $|T(x_i)|$, the vertical sum of the columns is the indegree $|T^{-1}(x_i)|$.

For a directed graph the adjacency matrix will not be symmetrical, however, for a non-directed graph with no loops, the matrix <u>is</u> symmetrical with all diagonal entries equal to zero.

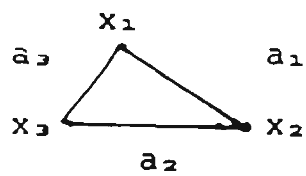
<u>Directed graph</u>.

$$A \quad \begin{array}{c|ccc|} & x_1 & x_2 & x_3 \\ \hline x_1 & 0 & 1 & 1 \\ x_2 & 1 & 1 & 1 \\ x_3 & 0 & 1 & 0 \\ \hline \end{array}$$



<u>Non-directed graph</u>.

$$A \quad \begin{array}{c|ccc|} & x_1 & x_2 & x_3 \\ \hline x_1 & 0 & 1 & 1 \\ x_2 & 1 & 0 & 1 \\ x_3 & 1 & 1 & 0 \\ \hline \end{array}$$



The <u>Incidence Matrix</u>, B is an alternative representation of T which gives a description of each arc.

$B = [b_{ij}]$, an n x m matrix where n = number of vertices and m = number of arcs in the graph. The element $b_{ij} = 1$, if x_i is the initial vertex of the arc, and $b_{ij} = -1$ if x_i is the final vertex of the arc.

Each column contains one 1 and one -1 entry, except that, by convention, the column for a loop has all zero entries. For non-directed graphs, all -1 become +1.

14. Important subset properties of a <u>non-directed</u> graph are :

A subset with the subgraph < S > complete - the <u>clique number</u> of G is the maximum cardinality of such a subgraph.

Subsets with < S > totally disconnected - the <u>independence number</u> of G is the maximum cardinality of such a subgraph.

Subset so that every vertex of X - S can be reached from a vertex of S by a single arc - the <u>dominance number</u> is the maximum cardinality of such a subgraph.

A <u>complementary graph</u> is formed by using the same vertices, joining all pairs of vertices that were disconnected in the original graph, and disconnecting all pairs of vertices that were joined. The clique number is equal to the independence number of the complementary graph.

$$\rho(G) = \alpha(\tilde{G}) \quad \text{and} \quad \rho(\tilde{G}) = \alpha(G)$$

15. A graph is <u>r-chromatic</u> if it can be colored with r colors so that no two adjacent vertices are of the same color. The smallest value of r is $\gamma(G)$ the <u>chromatic number</u> of the graph. Since no two vertices within a set can be adjacent, the coloring induces r subsets.

16. A non-directed <u>tree</u> is a connected graph of n vertices and n-1 links. A <u>spanning tree</u> is a partial graph of G

which forms a tree, ie (m-(n-1)) links of graph G are removed and the remaining partial graph is still connected. A graph of n vertices and m links may contain many spanning trees of the necessary n vertices and n-1 links. The <u>shortest spanning tree</u> connects the n vertices together minimising the total length.

17. The <u>shortest path</u> (in an arc-weighted graph) may be required from a specific starting vertex to a specific ending vertex.

A <u>Eulerian Circuit</u> in a non-directed graph, traverses every link of G once and only once. Such a graph contains zero, or two only, of vertices with odd degree.

If the links are weighted positive, a circuit which will traverse every link of G at least once with minimal total cost is known as the solution to the <u>Chinese Postman</u> problem.

A <u>Hamiltonian Circuit</u> is an elementary circuit passing once, and only once through every vertex of G.

For the arc-weighted graph, the problem of finding the circuit (or path) with the least total cost is called the <u>closed (or open) Travelling Salesman</u> problem. An alternative problem is to find the circuit (or path) whose longest arc is a minimum.

The open Travelling Salesman problem (shortest Hamiltonian path) is equal to the shortest spanning tree with the restriction that no vertex has degree greater than two. The solution to the closed Travelling Salesman problem is the shortest Hamiltonian circuit.

18. The above definitions and descriptions provide a basis for discussion related to the recording of data and methods of solution of problems in other chapters of this thesis.

5. GRAPH COLORING.

5.1. Introduction.

A vertex coloring of a non-directed graph without loops is required in the examination timetabling application. It is appropriate, therefore, to review the literature on exact and heuristic coloring algorithms, and on the reported performance of the algorithms on representative graphs.

The publications appear to cover a long time period because there has been a recent resurgence of interest in earlier work. Faster computers have made it possible to color larger graphs, which leads to more effective practical applications of the algorithms.

The method of presentation is to review briefly a large number of methods which have been proposed to find the chromatic number or to color a graph with the least colors, and then to refer to methods for coloring with restrictions.

5.2. Chromatic, Clique and Independence Numbers.

1. The chromatic number $\gamma(G)$ of a graph G is the smallest number r for which the graph is r-chromatic. Finding the chromatic number is equivalent to partitioning the vertices into r sets, each set containing only vertices colored with the same color.

The exact chromatic number may be difficult to find directly, but lower and upper bounds on it are more easily determined. According to de Werra (1975), " No formula has been found yet for the chromatic number of an arbitrary graph - thus most of the results in this area are bounds for $\gamma(G)$."

2. The <u>clique number</u> $\rho(G)$ of a graph G is the maximum number of vertices in a complete subgraph of G. As $\rho(G)$ colors are needed to color this subgraph, it follows that the size of the largest clique is a lower bound on the chromatic number.

$$\gamma(G) \geq \rho(G).$$

Corneil and Graham (1973) implemented this bound by finding an α-cluster (a set of a vertices containing a high density of edges) and adding edges until it became an α-clique.

Zykov (1952) showed that there is no further relation possible between $\gamma(G)$ and $\rho(G)$ and that the difference between the measures can be large. Tutte (1954) gave an example of a graph with 16 vertices for which $\rho(G) = 2$ and $\gamma(G) = 5$, however Dutton and Brigham (1981) colored this graph with four colors. Larson (1979) and Wagon (1980) showed special properties for graphs not containing K_4 or K_3 (see para. 4.2.10.).

3. The <u>independence number</u> $\alpha(G)$ is the cardinality of the largest totally disconnected subset of G, that is, the largest number of vertices that can be colored with any one color. Recalling that a maximal independent set of a graph is a clique of the complementary graph, it follows that :-

$$\gamma(G) \geq \lceil n/\alpha(G) \rceil \quad \text{and} \quad \gamma(G) \geq n/\gamma(\tilde{G})$$

where $\tilde{G}$ is the complementary graph
and $\lceil x \rceil$ is the largest integer less than or equal to x

Knowledge of the properties of the complementary graph may be of assistance in determining a lower bound on the chromatic number, as shown by Nordhaus and Gaddum (1956).

5.3. Bounds on the Chromatic Number.

1. A lower bound for the chromatic number was proposed by Geller (1970) :-

$$\gamma(G) \geq n^2 / (n^2-2m)$$

where n = the number of vertices, and m = number of links.

This bound is simple to calculate, and does not depend on knowledge of the clique number, the independence number, or of any property of the complementary graph. However, Myers and Lin (1972) showed that the bound $\gamma(G) \geq= \rho(G)$ dominates the bound of Geller, and suggested that, for some graphs, the lower bound of Geller may be considerably below the chromatic number.

2. An obvious upper bound due to Brooks (1941) is that of the maximum vertex degree plus 1,

$$\gamma(G) \leq \max_{x_i \in X}[d(x_i)] + 1$$

and one due to Nordhaus and Gaddum (1956) is, relating to the chromatic number of the complementary graph,

$$\gamma(G) \leq n - 1 - \gamma(\tilde{G}) \quad .$$

An upper bound given by Berge (1973) is

$$\gamma(G) \leq 1 + (2m(n-1)/n)^{1/2}$$

All these three upper bounds were shown to be too loose to be of practical significance by Christofides (1975).

3. Catlin (1978), extending the work of Brooks, showed that, if G is a graph with the maximum vertex degree >= 3, which does not contain a complete subgraph on the (maximum vertex degree + 1) vertices, then $\gamma(G) \leq$ the maximum vertex degree.

5.4. Exact Coloring Algorithms.

1. Exact algorithms, if carried to completion, guarantee an optimal coloring of the graph and the correct value for the chromatic number, but they are computationally difficult for graphs of real size. Exact algorithms will be surveyed under the headings of :- tree search algorithms, integer programming formulations, node reduction algorithms, and set-covering algorithms.

2. A simple implicit enumeration method to find the chromatic number due to Gillian (1970) and Brown (1972) is shown here as an example of a Tree search algorithm.

The vertices are ordered in any way, x_i being the i^{th} vertex.

(1) Color x_1 with color 1.

(2) Color each remaining vertex sequentially with the lowest possible color number (i.e. a color which has not been used for any vertex adjacent to x_i).

This produces a feasible coloring using q colors. Now attempt to produce a coloring using (q-1) colors.

Define x_{i*} the first vertex colored q. To produce a coloring of (q-1) colors, the vertex x_{i*} and at least one of its adjacent vertices must be recolored.

(3) Of the vertices which are adjacent to x_{i*} , define the one with the largest index as x_k (colored with j_k). Recolor x_k with the lowest numbered feasible alternative color $j'_k >= j_k + 1$.

(4) If $j'_k < q$, recolor the vertices x_{k+1} to x_n. If color q is not needed, then a new better coloring has been found.

(5) If color q <u>is</u> needed, then backtracking takes place either from the vertex requiring color q, or from x_k , until vertex x_1 is reached.

If the vertices are ordered in such a way that the first p vertices form the largest clique of the graph G, then, as these p vertices must by definition have separate colors, backtracking can stop when it reaches vertex x_p .

Christofides (1975) described the algorithm as a very primitive tree-search without bounds, but as being at least as good for graph coloring as any other available method. He also stated that a 30 vertex graph of 145 links could be optimally colored in 30 seconds using an IBM 1130 computer.

Korman (1977), in an attempt to improve on tree search algorithms, ran vertex-by-vertex coloring using successively fewer colors or successively greater colors. He also tried an iterative procedure of dichotomous search by which any two non-adjacent vertices are coalesced if of the same color, or an arc added if of a different color, but these procedures showed no improvement in computational efficiency. Taking a depth-first search rather than a breadth-first search more quickly established a feasible solution which can then be used for bounding purposes, and taking a restricted choice of sets to be colored did improve the performance of the algorithm.

McDiarmid (1979) found that the number of steps in certain branch-and-bound methods for determining the chromatic number of a graph grew faster than exponentially with the number of vertices in the graph. From the relatively recent results of Campers, Henkes and Leclercq (1987b), the practical experience is that the largest problem that can be solved is about 100 vertices.

3. In formulation as an <u>Integer Programming Problem</u> an upper bound q for the chromatic number is first found by any convenient approximate method. To find if a coloring of G using q colors is feasible, a matrix Z is built using vertices v_i, i=1,2,..n, against color r, r=1,2,..q, defining the element z_{ir} = 1 if $C(v_i)=r$, else = 0. The following set of conditions must be satisfied to obtain a feasible q-coloring of G :-

$L(1-z_{ir}) - {}_{j=1}^{k} a_{ij}z_{jr} >= 0$ i=1,2...n   r=1,2,...q

where L is a constant > n, and $A=[a_{ij}]$ represents the adjacency matrix of G. In addition, a formal objective function can be introduced so that no color r will be considered should colors 1 to (r-1) be sufficient.

The integer programming formulation involves nq constraints, and the size of the problem that can be solved depends on the capacity of the computer package. By enumerating a special class of finite mappings rather than the customary set of binary solution vectors, Marsten (1974) was able to solve problems of up to 200 nodes.

4. Zykov and Brunaldi's algorithm, as presented in Dutton and Brigham (1981), assumes a graph G with an initial node set $\{v_1, v_2, \ldots v_n\}$. Each iteration reduces the size of G by one node. Node v_i then represents the original node and all nodes merged with it.

Step 1. For all non-adjacent nodes v_i and v_j compute c_{ij}, the number of common adjacent nodes.

Step 2. If no non-adjacent nodes remain, stop, else determine the non-adjacent pair v_i and v_j with the highest c_{ij}.

Step 3. Merge v_i and v_j, adjust the c_{ij} value for all other non-adjacent pairs. Reduce n by 1 and go to step 2.

The preferred selection of pairs of nodes for merging should forestall the formation of a complete graph for as long as possible, by merging the pair with the maximum number of common adjacent nodes.

Dutton and Brigham (1981) concluded that this algorithm performed well on a wide variety of graphs as well as minimally coloring any bipartite graph.

5. In a solution as a Set Covering Problem, Christofides (1975) suggested the generation of all t maximal independent sets of the graph G as a matrix M,

where $M = [m_{ij}]$, and $m_{ij} = 1$ if vertex i is in the j^{th} independent set, and zero otherwise.

The coloring problem is then one of finding the least number of columns of M which cover all the rows. There may be several such column sets of equal size which are all solutions to the problem, and, having decided on one column set, it may be possible to interchange vertices between the columns by inspection.

To reduce computation, Korman (1977) considered only maximal independent sets of G as potential coloring sets. Srimani, Sinha and Chandury (1978) used the property that a maximal complete subgraph of the complementary graph is a maximal independent set of the vertices of G to provide a lower bound on the chromatic number of G.

5.5. Approximate Coloring Algorithms.

1. These methods find either an approximate chromatic number, or a feasible coloring of the graph, and are faster than complete enumeration methods. Christofides (1975), Carter (1986), Lotfi and Sarin (1986), and Campers, Henkes and Leclerq (1987), all summarised the methods available, and the last-named authors reported a large computational experience. The methods may be described as: sequential-by-vertex, sequential-by-color, saturation, bichromatic

interchange, coloring pairs, vertex removal, and complete subgraph methods.

2. The common principle in Sequential-by-vertex methods is to rank all vertices, say in decreasing order of degree, and then color them in order of rank, e.g. Welsh and Powell (1967), Wood (1969). A second order key may be used to separate vertices of equal first rank e.g. Williams (1974), and the remaining vertices may be re-ranked before the next coloring as in Carter (1978).

The ranking may be recalculated after first removing the vertices of small degree before coloring from the top, as done by Matula, Marble and Isaacson (1972). Recent numerical experience has been reported by Peemöller (1986).

3. In Sequential-by-Color heuristics the principle is to use one color fully before adding another. A method starting with the vertex of highest degree was proposed by Peck and Williams (1966). Methods based on Approximate Maximum Independent Sets AMIS were due to, among others, Leighton (1979), Chiba (1982) and Campers et al.(1987).

4. In the Saturation Method or Desatur Method, recorded by Brélaz (1979), the uncolored vertices were sorted, firstly, by the number of colors on adjacent colored vertices, and secondly by the degree in the uncolored subgraph.

5. In the Bichromatic Interchange Procedure of Campers et al.(1987), at each step of any of the vertex ordering methods, before creating a new color, an attempt may be made to interchange any color for which there is only one

conflicting vertex, thus allowing back-tracking, and improving the performance of the algorithm. According to Carter (1986), this procedure, combined with a "smallest degree last" sequential-by-vertex algorithm tends to use fewer colors than other methods.

6. Another method of speeding up the solution is based on Coloring Pairs, where the same color is given to the two vertices having the largest number of common adjacents. See Wood (1968,1969) and Campers et al.(1987).

7. By a Vertex Removal Method, due to Szerekes and Wilf (1978), all vertices of degree less than or equal to some value are removed from the graph, together with all edges incident on them, and the process repeated until no such vertices remain. If c is the smallest value for which the resulting graph is null, then (c + 1) is an upper bound for the chromatic number.

8. Campers et al.(1987) also refer to Complete Subgraph Methods whereby the coloring is started from a large complete subgraph, and uncolored vertices are added in order of those with the greatest number of prohibited colors.

9. The performance of heuristic methods may be ranked by either speed (CPU time), or by non-optimal performance (the number of excess colors used to color the graph). In tests on 80 graphs with up to 100 vertices and density 0.2 to 0.6, by 16 methods, Campers et al.(1987) found reasonable time-performance for all algorithms except complete sub-graph methods.

Adding interchanges to the original coloring increased the time taken, eg from 0.5 to 36.8 seconds, but reduced the number of colors used from 22.4 (average) to 21. The results were highly dependent on the kind of graphs. The RLF method of Leighton (1979), which selects a node at each step which will leave the remaining nodes colorable in as few colors as possible, showed the best results on average.

The performance guarantee (the worst case ratio between the number of colors used and the chromatic number of the graph) was studied by Wigderson (1983) and Bender and Wilf (1985). They offered proof but no practical implementation of an efficient polynomial-time algorithm with an improved performance guarantee.

In this author's work over the period 1982 to 1990, a great improvement in performance has come from faster hardware and software. Run-time has been reduced from 45 minutes to 4 seconds with very little change in the programs, and the whole suite of examination scheduling programs can be run on the micro-computer in less time than it used to take to log on to the main-frame computer.

5.6. Coloring with Restrictions.

A number of approaches to practical problems where there were restrictions to the coloring have been described in the literature. Relating to pre-assignments and restrictions to colors that could be used, Neufeld and Tartar (1974, 1975) showed that a graph with constraints reduces to a graph without constraints, the pre-assignments and restrictions merely being considered as part of the ranking order.

Scott (1976) and Mehta (1981) added extra edges and vertices to indicate the restrictions, and Leighton (1979) marked the vertices pre-assignment or priority. Colors may be clustered, or placed in certain orders during the coloring process according to Hell (1978) and Clementson and Elphick (1983). Salkin (1975) described a "Knapsack" constraint, whereby the sum of vertex weights on any vertex is subject to a maximum.

In conclusion, there are many algorithms available to color a graph. Whilst an exact algorithm will give an optimal solution, a heuristic, chosen to exploit the structure of the graph, may give a satisfactory solution to the problem in much less time.

6. THE TRAVELLING SALESMAN PROBLEM.

6.1. Introduction.

In the model for the examination timetable system, a solution to the Travelling Salesman problem is required. The heuristics and bounding procedures described in the literature are summarised below, using the terminology in the Graph Theory chapter 4.

Much of the discussion here is due to an excellent review and set of practical solution algorithms by Boyd, Pulleyblank and Cornuéjols (1988). Their recent publication postdates the commencement of the work reported in this thesis, but confirms that the algorithms adopted by the author are still performing adequately.

The method of presentation is to describe a number of methods which have been proposed to solve the Travelling Salesman problem, with comments by Boyd, and by other authors as appropriate, on the performance of the algorithms. For the small problem (20 nodes) encountered in the author's Examination Timetable system, any of the heuristic methods described are satisfactory,and searching for optimal methods is not worthwhile.

6.2. Tour Construction Methods.

1. In the Nearest Neighbour Heuristic a starting vertex is chosen at random, and a tour constructed , using at each

step the shortest arc (link) to an unvisited vertex. The path may return to the starting vertex (closed tour) or remain open. This is a particularly easy heuristic to program, but the later steps may be long when most of the options involving the remaining cities have already been used, and the method may be far from optimal, according to Reingold (1977).

2.In the Nearest Insertion Heuristic an initial tour joining two vertices very close together is expanded by inserting one vertex at a time. The vertex to be introduced at each stage is the one which increases the length of the tour by the least amount, and the method gives a tour which is, at most, twice the optimal length, again according to Reingold (1977).

3. For the Furthest Insertion Heuristic, the initial tour joins two vertices far apart, and the next vertex to be inserted is the one which increases the length of the current tour the most, but inserted in the cheapest way. Boyd considers this to be a useful general purpose heuristic.

4. In a quite different technique of a Space Filling Curve, N points in a square of area A may be sorted according to their images under a space-filling mapping to give a tour of length at most $2(NA)^{1/2}$. If the points are statistically independent under a smooth distribution, with N large, the tour was said to be roughly 25% longer than the optimum, by the originators, Bartholdi and Platzman (1982).

The author of this thesis obtained the following results on a practical problem :-

N = 80 points. A = 100 x 100 area

Theoretical length of path $2(NA)^{1/2} = 1788$

Length by space-filling = 1427

Length by nearest neighbour = 684

and concluded that the space-filling algorithm was inferior to the nearest neighbour algorithm, on the particular problem where the vertices were clustered.

5. Another available technique is the Sweep Heuristic in which an origin in the "centre" of the vertices is located, and the points sorted on the basis of their polar coordinates. The method guarantees a non-crossing tour but is difficult to visualise on a non-planar graph. However, it has been programmed by Boyd.

6. The Shrink and Cluster Original Travelling Salesman Heuristic (SCOTCH) originally described by Litke (1984), has been the basis of an experimental routine by Boyd. Vertices are grouped to form clusters so that the minimum distance between any pair of vertices in different clusters is maximised, and either furthest insertion or complete enumeration is used to form a tour.

7. In the method of Solution by Regions due to Karp (1975), each region is solved by an exhaustive search, and the small tours are linked by an algorithm similar to the greedy algorithm for finding an optimum spanning tree.

6.3. Tour Improvement Methods.

1. The tour constructed by any of the heuristic methods described above, can often be reduced in length by an improvement method according to Boyd.

2. In a Two Optimality Procedure, 2-opt, the procedure looks systematically at each pair of non-adjacent edges and, if the tour length is reduced, replaces them by the two alternative edges based on the same vertices. A similar procedure, 3-opt, may be followed by removing three edges.

3. In the Variable r-opt procedure due to Lin and Kernighan (1973), the interchange proceeds by removing an edge from the current tour, producing a Hamiltonian Path. Then one end of this path is joined to some internal node and an edge is removed, creating a new path - all internal nodes are tried. The "gain sum" is the sum of the length of the removed edges (except the last), less the sum of the lengths of the added edges. The procedure continues as long as the "gain sum" is positive and there are still unadded/unremoved edges.

All edges are considered for removal in turn, and the procedure attempts to maximise the difference between the length of the edge removed and the length of the edge added. The procedure as coded by Boyd takes a long time to run, and may be no improvement on the 3-opt algorithm.

6.4. Lower Bound Procedures.

1. The length of a Minimum Cost Spanning Tree is a lower bound on the length of a shortest tour. Prim's algorithm may be used to grow a tree from a starting node, adding the nearest node each time. Applying a post-optimality routine to a minimum cost spanning tree converts it to a tour which visits the nodes in the order which Prim's algorithm added them to the tree, and Boyd found it produced quite good results.

2. The Dual Ascent algorithm due to Lawler and others (1985) constructs a minimum cost spanning tree. Then for all degree one vertices, the second closest vertex in the rest of the tree is determined. The degree one vertex for which the distance is the greatest is joined to its second closest neighbour, creating a unicyclic graph called a 1-tree, the sum of the edge lengths being a lower bound on the length of a minimum cost tour.

3. Solving the Linear Programming relaxation of the travelling salesman problem provides a lower bound on the cost of an optimal tour. If the solution found is a tour, it is necessarily an optimal tour, according to Boyd (1988).

6.5. Branch and Bound Methods.

The description that follows of the general principles of branch and bound methods is due to Balas and Toth (1983).

The feasible set is broken up into successively smaller subsets; bounds are obtained by replacing the problem over a given subset with an easier (relaxed) problem; the procedure ends when each subset has either produced a feasible solution or has been shown to contain no better solution than the best already found.

Several different relaxation methods are available, all resulting in problems which have relatively simple integer programming solutions, eg the Assignment Problem, which may be solved by the Hungarian Method, or by using a restricted Lagrangean Dual. Results on problems of up to 325 vertices were reported by Balas and Toth, and ran in roughly polynomial time in respect to the number of vertices.

6.6. Performance of the Algorithms.

The practical work of Boyd et al.(1988) and Balas and Toth (1983), showed that the tour construction methods and improvement algorithms mentioned above can solve problems of practical size in reasonable time and with reasonable efficiency. The programs supplied on disc by Boyd allow the user to test the algorithms on particular problems, and to assess the performance.

The author has used the Nearest Neighbour Heuristic in the Examination Timetabling problem, as producing a quick and easy solution to the 20 node problem. Tour optimisation methods, with more complicated coding, show an improvement in quantitative terms, but these are overshadowed by such considerations as pre-assignments and the introduction of weekends into the schedule.

7. CLIQUE THEORY.

7.1. Introduction.

In Chapter 4 on Graph Theory, a clique was described as a complete subgraph. The size of the largest clique is a lower bound on the chromatic number of a graph, as every vertex in the largest clique must be colored with a different color. Alternatively, this may be expressed as the determination of the minimum number of sets required for the solution of the graph coloring problem.

The literature on cliques is based on three main topics - generating all the cliques in a graph, finding an upper bound on the clique number, and, finding the largest clique in a graph, - the last being the most useful in the timetabling application.

7.2. Generating all the Cliques in a Graph.

Several algorithms, based on the Bron-Kebosch Algorithm, for generating all the cliques of a simple, undirected graph, were described by Johnston (1976). The graph was defined either by an adjacency matrix or by a vector indicating the connected vertices, the latter being the more useful representation as the intersection of the vectors gave candidates for a clique.

In experiments, Johnston (1976) found that the domain of problems that could be solved in a reasonable time was either below 48 vertices with any density, or between about 100 vertices with 50% density, and 1000 vertices with 10% density. A result of interest was found in the experiment on relabelled graphs, where labling the vertices in ascending order of degree, or in descending order of degree, had little effect on the time of computation.

A tree-search algorithm for generating all the cliques of a graph was given, without experimental results, by Reingold (1977).

7.3. Bounds on the Clique Number.

A technique for refining an arbitrary upper bound for the size of the largest clique of a graph was presented by Selkow (1978). The limit on the upper bound was, initially, the degree of the vertex of largest degree plus one, and was reduced by considering the potential number of vertices which had a degree equal to the bound less one. The bound could be reduced further by partitioning the vertices into independent sets, and contracting each set to a single vertex.

A series of clique detection algorithms based on neighbourhoods in graphs commenced with the paper of Osteen and Tou (1973), who presented an approach which moved recursively from the given graph downward through a chain of subgraphs, removing one or more points at each progression.

A graph is "point-determining" if distinct vertices have distinct neighborhoods. The idea was used by Geoffroy and Sumner (1978) to establish an upper bound on the size of the largest clique in G in terms of the cardinality of the nucleus of the point-determinant of G (obtained by identifying points with the same neighborhood).

An upper bound based on the existence of a maximal cardinal clique that contained no vertex x, such that the neighborhood of x was contained in the neighborhood of another vertex y, was suggested by Billionnet (1981).

The concept of neighborhoods may be of value in finding the clique number in a large timetable graph of fairly low density.

7.4. Finding a Maximum Clique.

A new branch and bound procedure for finding the maximum clique in an arbitrary graph was described by Balas and Yu (1984). An algorithm for finding a maximum triangulated induced subgraph of G was extended to finding a maximum k-chromatic induced subgraph of G.

The algorithms were of $O(|V|+|E|)$ time complexity, and computational experience was given on randomly generated graphs of up to 400 vertices and 30,000 edges.

7.5. The Largest Clique.

The work of Carter (1984) was another approach to the problem of finding the largest clique in a graph. The basic method, due to Salazar and Oakford (1974), was to delete the vertex of minimum degree in a graph, until the remaining vertex set was completely connected. An upper bound on the clique number, given by the maximum degree of the deleted vertices, was found by Carter to be considerably overstated. By using a recursive version of the algorithm, in which each induced subgraph was examined for its largest clique, the upper bound was considerably reduced, the largest clique still being found in a reasonable time and in not more than five levels of recursion.

Attention was drawn by Carter to the difference between random graphs, in which the edges are randomly distributed, and typical timetable graphs in which vertices are related to each other by being part of prescribed courses or majors. The effect of this clustering in timetable graphs may be that the largest cliques are relatively small, and exist in separate clusters. A candidate for an eventual largest clique may be inadvertently removed at an early stage in the removal process, even though it is recognised in Carter's algorithm that removal of a vertex of a certain degree may remove a clique of that degree plus one.

7.6. Conclusion.

The problems of coloring the graph or of finding a largest clique are equally difficult. However, some algorithms have been described which have been successful on timetable type graphs, and which may be used to adjust the bounds on the chromatic or clique number found in other ways.

8. THE DSS FOR TIMETABLING AT FIT.

8.1. Introduction.

As indicated in the Chapter 1, five main areas of problems with timetabling were defined. Work was started with the object of improving the recording and reporting of data, and assessing the extent of the resources available and of the facilities required.

An approach of a Decision Support System, as described in Chapter 3, was taken. Such a system provides support, based on scientific management principles, for the decision maker. Good and feasible solutions, rather than optimum solutions are required.

Although at one stage an all-embracing management system was envisaged, circumstances dictated that five modular units were developed over a period of some years. The units were implemented at different times and under different jurisdictions. The units have operated together with the Institute's Student Record System, (itself changed considerably over the years), to provide record keeping, reporting and decision making support for the Faculty of Business and for the Institute Administration.

The system for examination timetabling has been in operation in the Faculty of Business continuously from 1982 to 1990, and the room scheduling system has served the Institute over the same period. The class timetable system has been used for the Faculty of Business timetables from

1983 to 1990. The student registration system has been used in several departments for some years. The examination recording system is of comparatively recent origin having first been used in 1988.

The systems are of varying degrees of sophistication and complexity. The examination timetabling system consists of a suite of programs in Basic language, and uses graph theory concepts to solve problems that cannot be handled manually. The class timetable system, on the other hand, is a series of decision rules, which, taken in order, produce a satisfactory timetable. The other three systems are data based, and have programs which facilitate the keying, recording and reporting of data, and assist with decision making.

8.2. The Overall System Described.

In Figure 1 will be found a diagram of the overall system, as it operates every semester. The five systems, shown down the middle of the page, are each described in detail in a subsequent chapter of this thesis. As a matter of policy, the construction of both the class and the examination timetables is done at Faculty or Department level, and the recording of room and examination timetables takes place at Institute level. Student enrolment and subsequent additions and withdrawals are in the institute's formal system, but may be supplemented at faculty or department level.

THE OVERALL SYSTEM FOR TIMETABLING AT F.I.T.

Planning) ______ Class Timetable Construction System — Ch.9.
Enrolments)
Room Availability)

Booked Classes) ______ Room Scheduling System — Ch.10.

Enrolments) ______ Examination Timetabling System — Ch.11.

Booked Examinations) ______ Examination Recording System — Ch.12.

Class Timetable) ______ Student Enrolment System — Ch.13.

Figure 1. The overall system for timetabling at F.I.T.

9. THE CLASS TIMETABLE CONSTRUCTION SYSTEM.

9.1. Introduction.

The production of a class timetable and room schedule for the Faculty of Business is a task that must be repeated each semester. The timetable is produced manually on a partly incremental and partly redeveloped basis.

Class timetabling is a two-round process. The first round timetable is produced before the student enrolment, which in turn determines the data for the second round timetable. The human decision maker tries to keep all the relevant constraints in mind, and to keep the differences between the two rounds to a minimum.

9.2. The Problem Described.

The Faculty of Business has about 2800 students, 100 academic staff, 14 courses, 150 subjects taught each semester, 28 classrooms and 20 dedicated rooms (1990 semester 2 figures). All subjects are semesterised, so a new timetable is required each semester. Enrolment for new students is at the start of the academic year in February, and for continuing students is in December for the next year. A mid-year registration is held in July, to update the enrolments for the second semester.

The requirement for the timetable is that it shall show: all scheduled student activities appropriately roomed, that the correct number of lecture streams and tutorial groups shall be provided for the enrolment in any subject, and that a viable timetable be provided for both full time and part time students.

Department and academic staff requests are considered, but it is not the responsibility of the Faculty Timetabling Officer to allocate staff to subjects, or to arrange teaching schedules. The normal group size for tutorials is between 16 and 20 students, for lectures it depends on the capacity of the room.

For first semester, new (intake) students are issued with a personal timetable, otherwise, students use the first round timetable as a basis for both subject and group enrolment.

The undergraduate courses in the Faculty have a common core of eight units, a compulsory specialisation of eight units, and an elective sequence of eight units (besides a year of Cooperative Education). The courses may be described as semi-structured, and course by year is far from a reliable basis for timetabling. The subject code is considered as the primary key for data management, followed by room and course. The major constraint on timetabling is actually the supply of rooms.

The data required as a basis for timetabling for a forthcoming semester is as follows :-

1. A list of subjects required by the departments.

2. Historical data of the enrolment of students in each subject.

3. Information on policy changes which may cause the enrolment to alter (e.g. intake student numbers).

4. For Round 2 - information on previous semester examination results which may cause the enrolment to alter.

5. A list of the classrooms and laboratories available.

6. A list of the students in each course, grouped into years.

7. A list of the compulsory, and desired elective, subjects in each year of each course.

8. Any special requests made by staff, or by groups of students.

9. Timetables and room schedules for the previous two semesters.

9.3. The Use of the System.

The timetable and the room schedule are built and recorded at the same time, so that it is not possible to timetable a class without rooming it, or to schedule two classes in the same room at the same time.

First, the number of enrolments in each subject, full time and part time, is estimated, and the number of groups calculated. This information is correlated with the staff hours available in the subject area.

Second, the lecture timetable is constructed and roomed.

Third, the classes in special use laboratories are allocated. Some of these laboratories are in use almost every hour of the week.

Fourth, the seminar subjects are allocated and roomed.

Fifth, the tutorials are allocated and roomed.

Then, a check is made for each subject area that there are not too many classes at the same time for the number of teaching staff available, and for each year of each course that students who are reasonably "on-stream" have a feasible timetable.

The first round timetable, after a short period for comment by staff, is printed for the student enrolment sessions. For first semester, a second round timetable, with amendments, is also printed, but for second semester, the amendments are written by hand onto the master timetable. Subject code and name, lecture, seminar, tutorial or practical session, and group, day, time and room are recorded.

9.4. The Student Enrolment Sessions.

Student enrolment is at arena sessions, attended by course/year groups of students. The policy is to allow a student to enrol in any subject desired, provided that the loadings of four subjects, full time, or two subjects, part time, are not exceeded, the necessary pre-requisite subjects have been passed, and the appropriate course outline is followed.

Students are checked into the desired subject and group by a tally procedure, advised to choose another group when one is full, and, as a last resort, put on a waiting list for the subject. Following the enrolment sessions, any necessary adjustments to the number of groups in a subject, and to the timetable, are made, detailed tutorial lists are printed, and the responsibility for each subject is handed over to the lecturer in charge.

9.5. The Success of the System.

Certain conventions regarding room bookings and class times have been established to aid scheduling, for example, all classes of two hour duration start on the odd hour. Generally, the process proceeds without incident, however, there are certain subjects which are changed almost every semester between the first and second round timetables, or even after the start of the semester, with consequential changes in student enrolment. Perhaps there is a need for more education of staff and students to get the timetable and the enrolments correct in the first place.

The system feeds into the Institute Room Scheduling System without any problems. The procedure of manual construction and computerised recording has produced feasible and useable timetables and room schedules in the Faculty of Business over a long period of time.

10. THE ROOM SCHEDULING SYSTEM.

10.1. Introduction.

The recording and analysis of the classroom schedule for the Institute is a structured task which must be done every semester. A recording system is required which is easy to use, and in which it is easy to alter the data.

An on-line interactive system, accessible to department timetable officers by terminal from the main-frame computer, was designed and implemented, but subsequently changed to a system on a microcomputer, accessible only to the Institute Timetable Officer. At the time of writing, there is a move to revert to an interactive system, as a result of changes to hardware availability and to the organisational base.

10.2. The Problem Described.

In 1982 the recording of classroom booking data was a part-time job for an academic staff member. Each department did their own room allocation, and the Institute Timetable Committee was charged with the task of settling conflicting claims to rooms.

When increasing student enrolments put pressure on space, an Institute Timetable Officer was assigned to the job. At the same time, an audit of space revealed that there were in the Institute about 180 classrooms, of which about half were "general classrooms", and half were "dedicated space, including laboratories".

A system was set up to record the room usage on computer in standard format, and departments were asked to submit their room requirements in advance of the start of the next semester. The procedure was then monitored week by week.

The usage or occupancy rate of rooms varied widely between buildings and between departments. Some rooms were used almost all of the hours of the week, many were used for about 20 hours a week, and some were used only a few hours. For other rooms, no data was provided by departments.

The computer file was revised every Friday afternoon, printed out, and delivered to the departments. Large numbers of additions and changes to room usage were recorded after the start of the semester, and continued right up to week seven (i.e. mid semester). It was clear that rooms must be pre-allocated in some way so that Department Timetable Officers could arrange classes before the start of the semester, and, without the need for discussion with the Institute Officer over each requirement.

10.3. The Allocation of Rooms.

In 1984, the Institute Timetable Committee took over the task of pre-allocating rooms. The four teaching faculties were taken as the base units, and the total teaching hours on campus of academic staff in each faculty as a surrogate for the classroom hours required. Each laboratory or dedicated space "owned" by a department or faculty was counted as 0.4 of a room, and the general classrooms were then divided up among the faculties.

The geographic location and sizes of the rooms were taken into consideration, the aim being to provide each faculty with a "fair" share of the good and the poor rooms. The allocation is revised yearly to take account of changes in room availability. Although not entirely satisfactory, the scheme does allow Departmental Timetable Officers to book rooms with reasonable certainty and has reduced the work load on the Institute Officer.

10.4. The Practical Use of the Programs.

The initial programs on the main-frame computer were replaced in 1987 by a suite of programs in dBASE III on the microcomputer. The programs facilitate data entry, provide hard copy room schedules three times a semester, and analyse and report on room occupancy rates. The data is provided by the departments, the programs run by the Institute Officer, and the hard copy returned to the departments for checking and for information.

Late requirements for rooms, and one-off bookings, are dealt with between the departments and the Institute Officer, mostly by telephone.

A User Manual for the system was provided by Southwell (1988).

10.5. Success of the System.

The main purpose of the system is to record room bookings, so that staff and students may be assured of a room for each scheduled class. In this respect the system has proved entirely satisfactory over 12 semesters to date.

A secondary purpose is to provide information to the Security Officer to allow for access at appropriate times, and to the Maintenance Staff to allow for maintenance of the rooms. The hard copy listings of room bookings are satisfactory for this purpose. The feature program ROOMDOOR, which prepared a hard copy of all bookings for the week for a particular room, is no longer used, being more trouble than it was worth (the number of changes to room bookings after the start of the semester, though much reduced, is still significant).

A third purpose is to analyse room occupancy rates. The reports generated by the system are valuable as a management tool, being used both within the Institute and for submission to the Department of Education and Training.

The Room Scheduling System is an exercise in data recording and in management organisation. The system serves an on-going purpose at Footscray Institute of Technology.

11. THE EXAMINATION TIMETABLING SYSTEM.

11.1. Introduction.

The construction of an examination timetable is computationally difficult, as many constraints have to be met. Identifying the constraints, and determining which of them are important, is also difficult, particularly if a tight time frame for the examination period, or for the construction of the timetable, applies.

The concept of a decision support system for the construction of the timetable provides for a preconceived framework for the process and a suite of prewritten programs for the computations, yet allows the application of human experience to the final decisions.

A Decision Support System for Examination Timetabling is presented here. The system is original work, and has been programmed, tested, implemented and maintained by the author over a period from 1983 to 1990 in the Faculty of Business at Footscray Institute of Technology. The programs in the system are described, both as to the mathematical models involved and as to the practical implementation. Further details of the coding may be obtained from the author.

Two Data Theory concepts, the searching of a list and the building of a data matrix, and two Graph Theory concepts, graph coloring and the Travelling Salesman algorithm, are used in the Examination Timetabling System.

11.2. The Problem Described.

A typical enrolment in the Faculty of Business is some 2700 students, made up of undergraduate students in seven faculty courses, graduate students in eight faculty courses, and "out of faculty" students doing some subjects in the faculty. An undergraduate course is of four years duration, and a graduate course is of two years duration. These rules, although subject to some exceptions, will suffice for the definition of the problem.

Each full time student does four subjects a semester, each part time student does two subjects, and most subjects had a formal final examination. An undergraduate course consists of sixteen compulsory subjects and eight elective subjects. As a subject failed is repeated the next semester, and an elective subject in one course may be a compulsory subject in a different year of another course, the resulting student enrolment can best be described as semi-structured, having some recognisable form, but no consistency in detail from one semester to the next.

The problem is to construct an examination timetable for all Faculty of Business subjects that will meet all the constraints and requirements listed below :-

1. All examinations are in the afternoon session (and/or evening session, if permitted).
2. The examination period fits in the allowed maximum (typically 15 to 18 days).

3. The timetable is clash-free, no student is required to sit two examinations on the same day.

4. All candidates in a subject, and in any subject with a common paper, are examined at the same time.

5. A maximum capacity constraint of seats at a session is met (currently 540), except where a subject exceeds this number of candidates.

6. A maximum number of subjects is examined at any one session (currently 20).

7. All subjects are treated as examinable unless the subject department notifies otherwise.

8. The number of candidates sitting on consecutive days is minimised.

9. Subjects with large enrolments are put early in the examination period to allow time for marking.

10. A limited number of requests from academic staff for particular dates are considered.

11.3. The Suite of Programs.

1. General. The programs are written in Microsoft Compiled Basic, and are located on the hard disc of an IBM PS/2 Model 50 computer. The student data file is obtained as an extract from the Institute's student data system at week 8 of the semester, supplied as an ASCII file on floppy disc. For their main purpose, the programs are run once at week 8.

2. <u>The Student Data File.</u> There is one record for each student who is enrolled in any subject in the Faculty of Business, giving the Student ID (7ch), course code (5ch), and the subjects enrolled in the Faculty (5ch each) e.g.

8900359 BFBF2 AE232 AL200 AL251 HA008

8903109 MADF2 AE125 AL108

(A space has been inserted between each field for clarity).

3. <u>The Data Cleaning Program.</u> The program allows a preview of the data. It prepares a list in alphanumeric order of all subjects found on the data file, and counts the number of students in each subject. After inspection of the list by eye, any subjects which are in error or not required may be removed from the data file, or marked for deletion of the examination at a later stage, as considered expedient. The program handles any number of records and includes a visual (screen) count of progress. Total run time is about 15 seconds for 2700 records.

A simple list-build algorithm (Knuth 1973), is used to identify the subject codes and to count the occurrences.

Step 1. Suppose N1 = total number of subjects on file,

N2 = maximum number of subjects per student,

N3 = number of records (students) on file.

Step 2. Read first (next) record.

Step 3. Split record into N2 subjects

Step 4. Read first (next) subject

Step 5. Find subject code on built list.

Step 6. If not on list, place subject code at the bottom.

Step 7. Add one to number of students in subject.

Step 8. If not last subject, go to Step 4.

Step 9. If not last record, go to Step 2.

Step 10. Record values of N1 and N3.

Step 11. Sort the list on key of subject code, and print.

The subject codes of the list are a set of vertices of a graph, the number of students in a subject are the vertex loadings, and the vertices are not, at this stage, linked.

4. The Matrix Build Program. The matrix build program takes the student data file, and builds a list of subjects, counting, for each subject : the number of students, the number of other subjects being done by the students (**subclash**), and the number of other papers being written by the students (**stuclash**).

The program also stores a clash matrix, of the number of **clashes** between any pair of subjects, representing the weighting on the links.

The algorithm of the data cleaning program was extended at Step 7 to record each combination of the subject with other subjects in the same record (**clash**), and at Step 10 to count the non-zero values in each row of the clash matrix (**subclash**), and total the entries in the row (**stuclash**).

Matrix M can be described as an adaptation of an adjacency matrix (Section 4.2), an N1 x N1 matrix where $M_{ij} = 0$ where $i = j$, $M_{ij} = 0$ where the vertices are adjacent, otherwise the value of M_{ij} is the arc-weighting.

5. <u>The Graph Coloring Program.</u> The program uses the clash matrix as data to allocate subjects to days in such a way that there is no clash between any two subjects on the same day, and the maximum capacity constraint is not exceeded. The allocation is done three times, in different vertex order. Also a "dayclash" matrix, the number of clashes between all subjects on any one day and all subjects on any other specified day, is stored for each allocation.

A vertex coloring of the non-directed graph without loops identified by the clash matrix of the previous program is required. A sequential-by-vertex heuristic (Section 5.5) proved to be adequate for the task, producing feasible solutions to the practical problems consisting of 150 vertices in a few seconds of run time.

The heuristic used is as follows :-

Step 1. Set current color number = zero.

Step 2. Take next ranked vertex.

Step 3. Add 1 to color number.

Step 4. If the current vertex is adjacent to any vertex already with that color number, go to step 3.

Step 5. Give to the current vertex the current color number.

Step 6. If not the last vertex, go to step 1.

In the practical application, some restrictions were placed on the coloring procedure to simplify the later decision process. As stated in Section 5.6, the restrictions are considered as part of the ranking order.

6. <u>The Travelling Salesman Program.</u> To minimise the number of consecutive day examinations, it remains to put the days in the best date order. The data is the "dayclash" matrix produced in the previous program. A nearest neighbour heuristic, as described in Section 6.2, provides a convenient solution to the small problem.

The procedure used is as follows :-

Step 1. Set upper bound distance to large.

Step 2. Start from first (next) vertex. Set trip distance to zero.

Step 3. Select nearest unvisited vertex.

Step 4. Move to that vertex. Add arc-weight to trip distance.

Step 5. If not last vertex of trip, go to step 3.

Step 6. Replace upper bound distance with trip distance if the latter is smaller. Record details of trip.

Step 7. If not last starting vertex, go to step 2.

7. <u>The Output from the Programs.</u> The decision maker now has available three alternative listings, with different combinations of subjects suggested for each day.

8. <u>The Clash Check Program.</u> A program to determine the number of potential clashes between any pair of subjects is available.

11.4. Case Study for 1990 Semester 2.

To illustrate the Examination Timetabling System, the actual process is given for 1990 Semester 2 for the Faculty of Business at Footscray Institute of Technology.

Table 1 shows that there were 2706 students sitting a total of 6598 papers in 147 subjects. Seventeen sessions were available for the examinations, with a maximum capacity of 540 students at each session.

At Week 8 of the semester, the programs were run, and the Draft Examination Timetable issued. At Week 10 of the semester, the Final Examination Timetable was issued, and the data passed on for inclusion in the Institute Examination Timetable.

A list of the files and programs is given in Appendix A. A copy of the Basic language code of the programs is available from the author.

DATA FOR THE SEMESTER 2, 1990 EXAMINATIONS

STUDENT DATA

Department	Code	Subjects	Enrol
Applied Economics	AE	52	2268
Accountancy & Law	AL	35	2390
Faculty - Cooperative Education	FB	5	202
Faculty - Masters	FB	14	133
Hospitality & Tourism Management	HA	41	1605
Faculty of Business students 2706		147	6598

EXAMINATION DATA

Program	Sessions	Subjects	Students
Undergraduate	17	102	2522
Masters	6	14	51
Graduate Diploma	5	26	133
Cooperative Education	0	5	included in undergraduate
		147	2706

TABLE 1. Data for the 1990 Semester 2 Examinations.

An example of part of the output from the matrix build program is :-

Subject	Students	Subclash	Stuclash	Index
AE006	431	26	1092	15
AL213	161	53	420	86
HA181	72	21	206	6

(The complete list of subjects is given in Appendix B). The meaning is that, for subject AE006, there were 431 students, doing 26 other subjects between them, and writing 1092 other papers. For subject AL213, the 161 students were doing 53 other subjects and writing 420 other papers. This suggests that 54 sessions would be required to avoid clashes.

In the Graph Coloring Program, the vertices were colored in three orders, giving three sets of output, as shown in Appendix C. First, the allocation was done in FIFO order, just as the data happened to be recorded originally. Second, the allocation was in decreasing subclash order, and, third, in decreasing stuclash order. The second allocation, shown in Appendix C2, will be described in some detail.

The allocation used 15 sessions, placing 16 subjects on the first day, occupying the available 540 seats, 6 subjects on day 2 again 540 seats, and so on until there was one subject left at session 15 with 19 seats. The controlling constraint at six sessions was the available seats, at one session the limit of 20 subjects, and at eight sessions the need to avoid clashes. The next step was to adjust the order of the sessions to minimise the number of consecutive day examinations. This is illustrated using the allocation in decreasing **subclash** order in Appendix D. The best route shown by the Travelling Salesman program is day 4 followed by day 15 followed by day 2 etc., for a total of 537 consecutive day examinations.

At this stage some human decision making was applied to introduce weekends and to meet specific requests for certain subjects on certain dates. The resulting Draft Examination Timetable is shown in Appendix E. A few further requests for changes were checked out using the Clash Check program. The working time to produce the Examination Timetable, given the programs and the data, was about one hour, plus about four hours to deal with the late requests for changes.

11.5. The Success of the System.

A suite of programs has been developed by which an examination timetable is worked out each semester. The system has been implemented in the Faculty of Business, and has been maintained for eight years.

The examination timetables produced by the system have always been free from clashes, and are considered to be satisfactory by staff, students and administration. They are easy to feed into the next stage Examination Recording System (Chapter 12).

The complete success of the system in the Faculty of Business has not however, been matched throughout the Institute. The system was considered to interfere with the power of departments to set examination dates. If the objection is withdrawn in future, the system could be used in individual departments without difficulty, but would need some revision for use by the whole Institute at one time.

12. THE EXAMINATION RECORDING SYSTEM.

12.1. Introduction.

The recording and publication of the examination timetable is a structured task which is done repetitively every semester. It is not a task of great mathematical complexity, nor does it require expert decision making capabilities. Nevertheless, it can gain in accuracy, ease and speed by the application of simple data base techniques and a DSS.

12.2. The Problem Described.

The Examinations Officer at Footscray Institute of Technology receives, at the end of week 10 of the 14 week semester, from each department, a list of the subjects, and of the dates, for formal examinations. Upon assembling the requests, two main problems are encountered - the total seating capacity will be exceeded at some sessions - and there are a number of clashes, that is candidates are required to sit two subjects at the same time.

The original manual procedure was cumbersome, involving writing and checking subject codes, names and dates. The production of a clash report was a relatively efficient computerised procedure, but it was rendered ineffectual by the many additions, deletions and date changes submitted by departments right up to the date of the examinations.

The Examinations Officer administers the production of detailed candidate lists (from the computer), and of detailed session schedules (produced manually).

12.3. The Practical Use of the Programs.

A suite of programs written in dBaseIII on the microcomputer is used to facilitate data entry, calculate numbers of candidates at each session, facilitate data changes, and to report the list of examinations in subject code and in date orders. Computing Services provide a master data list of subject codes and names for use on the microcomputer, and continue to process the clash report on the main-frame computer.

The Examinations Officer makes the decisions about which subjects to move when the total seating capacity is exceeded. Some support is provided for the room allocation and for the production of the documentation used in the examination centres.

12.4. The Data Base Model.

The model is based on a small relational data base of three files, the list of examinations, the list of subject codes and names, and the list of timeslots. The files are indexed to allow ready access, and the programs, seven in number, produce first an expanded version of the examination file, and then various reports on candidate numbers, and subject codes, names and dates.

12.5. Success of the System.

The system was implemented at Footscray Institute of Technology in 1988, and has been in use since then. Documentation has been produced on an internal basis, as it is used by only one officer.

The examination recording system may be considered trivial in nature, however it follows in its design the precepts of a Decision Support System, and is a valuable support system for management. The result is a system which improves accuracy, saves time and work, and which the Examinations Officer actually enjoys using.

13. THE STUDENT ENROLMENT SYSTEM.

13.1. Introduction.

A simple student record system was required for use in the Faculty of Business, to supplement the Institute formal system. The system allows faculty administrative or academic staff to enter more detailed records on each student than are needed under the Institute system, and to have access to the current detailed records without security risk to the Institute system.

Common data structures with the class timetable and the examination timetable programs are used. The data for the system is gathered at the student enrolment sessions. Reports on enrolment by student, by group, by subject and by course can be produced.

13.2. The Problem Described.

The courses in the Faculty of Business are semi-structured, many combinations of subjects and groups are possible. It is necessary to know, for both planning and operational purposes, which subjects and groups a student is doing. The Institute Student Record System includes in the student's academic history a list of the subjects being studied, but not the groups. The Institute system also lists the students in each subject, and in each course, but the information is not available until several weeks into the semester, too late for planning purposes. A system was therefore set up within the Faculty of Business to provide for all the needs mentioned above.

13.3. The Use of the Programs.

A data base structure was set up to store the student ID, name, course code, and, for each subject, the subject code and group number. An identical structure is used in the class and examination timetables, the room schedules, and the faculty enrolment procedures, so that data is entered only once.

A menu driven suite of programs in dBase III allows users to make inquiries about individual students, tutorial groups, subjects, courses, or department enrolments. Lists can be printed in alphabetical name order or numeric ID order, as described by Southwell (1988b).

13.4. The Success of the System.

The system operated in the Faculty of Business from 1983 to 1988. It was then declared redundant, because of improvements in timing in the Institute system, and because the increased number of students had made the keying of data a tedious chore. However, in 1991, the system is to be reinstated at the request of the subject lecturers.

The system was also installed in the Departments of Mechanical Engineering, Chemistry and Biology, and Humanities.

14. CURRENT EXAMPLES OF DSS FOR TIMETABLING.

14.1. Introduction.

A number of site visits were made for research discussions with the authors of papers mentioned in Chapter 2, to determine the current status of the systems. It soon became evident that no two systems are alike in policy or in extent, and that all systems contain interesting specific aspects.

More work on surveying the extent of the use of computerised scheduling, on determining the appropriate design factors, and on considering the factors which lead to successful implementation is outside the scope of this thesis. All that can be done here is to describe each system briefly.

14.2. University of Waterloo.

The algorithms for the class and room scheduling system at Waterloo were written by Michael W. Carter. The system was installed in 1987, and is now controlled and run by David Mason of the Student Course Project Group and Janice Foster of the Data Processing Department.

The system provides a complete timetable for the university, with associated room schedules, staff timetables, and student timetables. It is linked to the student registration system, and provides for student billing, correspondence student course guides and book lists. It does not do examination timetabling. It runs on the mainframe computer, which has both staff and student access allowed at certain times and for certain purposes, and which prints several reports.

The first assignment run is made three months preceding the university term using the timetable already generated manually in the departments, the student pre-registrations (already deposited), and the room inventory file (already decided). Following the report on the first assignment, some alterations are made to the timetable to make it meet the room, department, and student registration requirements.

At the second assignment run, each student is assigned to a section of each course, and is given a feasible timetable. When "marks processing" for the previous term is run, it puts out warnings on changes of courses needed for next term (due to failures), and the student may then change the course or section, working interactively on the computer.

The whole system of student registration, timetabling, and room scheduling consists of 1300 computer programs and has as many as eight staff members working on it at certain times of the year.

14.3. State University of New York (SUNY) at Buffalo.

The algorithms were written by Vahid Lotfi,the system was implemented over a period of some years, and, in 1989 was taken over and run by the University Administration.

The multi-phase final examination scheduling process consists of first grouping the final exams into sets called blocks, in such a way as to minimise the number of students with simultaneous exams. Second, the blocks are assigned to exam days, while minimising the number of students with two or more exams a day. Third, the exam days and exam blocks within days are arranged so as to minimise the number of students with consecutive exams. Fourth, exams are assigned to classrooms so as to maximise space utilisation.

The examination period at SUNY is of only one week's duration, and the times follow, roughly, the class timetable, (a manually produced timetable which is not part of the system). Policy calls for two seats per candidate, and requires that the sections of a multi-section course are examined together in some, but not all, cases. The constraints on the number of sessions for examinations are very tight under this system and some clashes occur which are later sorted out by the students and the instructors.

14.4. University of California at Berkeley.

The programs at Berkeley are run on several computer systems in several languages, only the two major systems will be described. The Room Scheduling System, written by Roger Glassey, was implemented in 1987, and is now run by LaVern Lazzereschi, Classroom Manager, Office of Admissions and Records. Sadishev Adiga, a colleague of Roger Glassey in the Department of Industrial Engineering and Operations

Research, told of a interesting recent development in the reprogramming of some of the system from linear programming to expert systems in Prolog.

The class timetable is produced by the departments and roomed by the Classroom Manager, who has the right to request a change of time for a class that cannot be roomed. Standard time patterns are used for courses/rooms, thus there are a limited number of permitted combinations.

Final examinations are run in an eight day period, a total of 22 sessions. Some, but not all, multi-section subjects are examined at a common time. The class and draft examination timetables are available at the time of registration, and it is up to students to select subjects not to clash, however there is considerable effort involved by Room Scheduling to finalise the examination timetable.

The Classroom Manager also records once off and evening classroom bookings on a daily basis, and confirms them in writing to the department concerned and to Custodial Services.

The second system at Berkeley is the Advance Class Enrolment (ACE) System, written over a long period of time by several authors, implemented in 1987, and explained to me by Russell Low, Analyst, Office of Admissions and Records. Students collect registration papers at an arena session, then submit course requests on mark-sense forms by an appropriate deadline.

The course request forms are scanned into the computer, and the first stage of the ACE run is a demand analysis. As some courses are much overapplied, the "older" students are given priority, but the timetable, driven by supply rather than demand, is not altered.

As the second stage of ACE, the computer system, (run in batch mode), schedules the students into classes, and issues to each student an official class schedule shortly before the start of the semester. An add/drop process applies for the next two weeks.

14.5. Memorial University of Newfoundland.

The programs for the system at Memorial were written and implemented by Gary Sabin, Faculty of Engineering and Applied Science, and are run by Joseph Byrne, Assistant Registrar/ Admissions Manager.

A class timetable for the coming three semesters is issued two months prior to the start of each semester. The central enrolment system records details for each student, course and section. Two separate methods of access to the central registration system operate at Memorial, for re-enrolling students the telephone registration system, and for new students the in-person registration.

For re-enrolling students, the telephone registration system is available from two months before, until the week before, the start of semester. An add/drop facility is available in the first two weeks of semester. The TRS is accessible by touch-tone phone, and has a voice response system.

All full-time students in their first semester at Memorial enrol in the Faculty of General Studies. They attend for interview, on the Tuesday prior to the start of semester, and apply for the courses required for their intended eventual major study. A first computer run is done that night to assess the numbers, and a meeting held during the next day (of the course leaders) to adjust the timetable. A second computer run assigns to every student a feasible and clash-free timetable. Personal timetables for students, and class lists for instructors, are available on the Friday before the start of semester.

Every effort is made to provide the courses that students want, with balanced section sizes and reasonable timetables. The allocation of sections is on an "oldest" first basis for re-enrolling students, and on a random order basis for new students.

14.6. University of Colorado at Boulder.

At Boulder, new student admissions are done well in advance, and, after interview, the students join in the normal student registration scheme. A revised centralised and fully computerised scheme commenced in 1987. Bob Preston is the Associate Registrar in charge.

Timetable data is submitted by the departments. Assistant Dean Karen Bever and Richard Nishikawa of the

College of Arts and Sciences, told me that their timetable was based on the resources available, and was built by a process of roll-forward with specific adjustments.

Central "Course Inventory" and "Room Inventory" files are kept by Susan Boehme of Academic Room Scheduling. On receipt of a request to hold a class, the computerised system will select the "best" room available, or will indicate the best alternative time and room for the class.

Each student receives a written "Invitation to Register" about two months before the start of semester. The student may need advising first, (eg. if the Grade Point Average to date is low), and then calls CU Connect on a touch phone to register for courses and sections. CU Connect searches for an open section if the wanted section is full or has been cancelled. Wait lists are provided for in the situation that all sections are full or the student will not accept any of the open sections. The wait lists are circulated to the departments weekly. The student gets an immediate confirmation of registration, and may call CU Connect again and add or drop courses or alter sections.

The examination timetable is determined by the normal weekly class times. The period is one week including Saturday, and different sections of the same course are examined at different times. This contrasts with the policy at Footscray Institute of Technology (see Chapter 11.2, para.4).

14.7. University of Waikato.

John Buchanan of the Department of Management, is the (two year rotating) Chairman of the Timetabling Committee. Most timetabling is done manually. Specific problems evident at Waikato are the lack of teaching space, and the inability to forecast student numbers and class sizes.

Teaching space has a usage rate of over 95%. The bookings are kept on computer, and displayed on a huge wall board in the Room Scheduling Office. Kathy French is the Room Scheduling Officer.

The timetable for two semesters is published in advance, with corrections given at enrolment time. Late alterations cause problems with rooming. The policy is to admit all students who wish to attend the university, and to register them for all courses that they wish to do, hence the numbers in classes are changing right up to week 4 of semester. Up to 10% clashes are acceptable, and the students must then sort out their own problems.

14.8. University of Nottingham.

The following information was provided in a letter from Mrs B.Hicking, the Examinations Officer, University of Nottingham (1986).

A computer operated examinations timetable, originally written by a research student in the Mathematics Department, has been in operation since 1970. The first, computer, phase produces a clash list between subjects. The second, computer, phase allocates the subjects to sessions, permitting 1400 seats a session and allowing priorities referred to as one examination per day, very early in the period, and large.

A third, manual, phase allows for minor adjustments to the schedule, and around 850 papers are scheduled in 26 sessions. The system has recently been revised to suit the newer computers.

14.9. General Comments. Comments on timetabling systems are recorded verbatim, and confirm the observations in Chapter 1.8.

Gary Sabin (Memorial) :- "The system requires the full cooperation and support of management. It then works to the advantage of management, staff and students, and particularly helps new students to integrate to university life."

Vahid Lotfi (SUNY) :- "A system should utilise procedures which can be solved in a reasonable time frame without requiring excessive computer resources. There is a trade off between the timeliness of the solution, the ease of implementation and the degree of exactness required. The results of the efforts (in introducing the system) help to improve the quality of university life by allowing proper planning and reducing frustrations."

Bill Hathaway-Clark (Boulder) :- "The system must cope with the processing of students for admission which goes on all year, and with subject registration with its frequent add/drops."

Ernie Hudson (Berkeley) :- "The funds are carved up between departments on the basis of student full-time-equivalents, but the students' preferences for majors are not considered in setting the timetable."

15. CONCLUSION.

15.1. Discussion.

Most, if not all, universities, have computerised student record systems, but very few have computerised timetabling systems. None, as far as I know, has a fully computerised, fully integrated, academic record system, covering student records, registration, and timetabling for staff, students and rooms for both classes and examinations, (Waterloo is the nearest).

The quantitative methods appear to be available, as do the necessary tools for the design and implementation of systems. Many case studies appear in the literature of the development of a system for, say, room scheduling, without the system having been implemented.

From the description of current systems in Chapter 14 it is evident that to implement a system takes a long time span. The author of the system must have some technical knowledge, some managerial ability, and most important, a high degree of persistence. The author must be encouraged by higher academic management to implement the system, and given the enthusiasm of middle management, academic staff and computing services staff to complete the changeover from a manual to a computerised system. The task is much bigger than it appears at first sight.

However, the task is more manageable at a lower level with smaller systems or sub-systems which do not involve so many people. If the systems fill a need, they will be used. It is worth splitting plans for large systems into smaller sub-systems which are manageable and can be implemented without opposition. At a better time, practically or politically, more of the large system may be implemented.

The fashion in universities that each staff member (and in some places, each student), designs his or her own timetable does not agree with constraints on resources. Other problems are; a very uncertain or fluid level of student registrations and a great shortage of teaching space.

A new move in the USA and Canada towards phone-in registrations extends the registration period, and allows the student more freedom in selecting subjects and times. But it still requires a class timetable to be published before a student can select classes, and, the registration to be completed before the timetable can be finalised. There is no way out of this dilemna. Either the timetable must be done twice, or the registration must be done twice.

If the registration is known, the timetable can be written with certainty, this being the situation for examination timetabling and for some class timetabling in the school situation. It is important then to be able to allocate resources in a general way, and not to identify too many items as "special". It is also important to have some flexibility in the allocation, without having to waste time by doing it twice, or make mistakes by doing alterations.

Many methods are available for the quantitative allocation of resources, and the method used depends on the experience of the author of the system, and the hardware and software available to the user. Both the method and the implementation must be flexible enough to adjust to minor changes. Also there must be regular maintenance to adjust the method to changes in associated systems.

15.2. A Comparison of the FIT and other systems.

The point was made in Section 14.1 that " no two systems [reviewed] are alike in policy or in extent, and that all systems contain interesting specific aspects." A brief comparison along these lines follows.

The Southwell System at FIT is an adjunct to the Institute's Student Record System, and not part of the official policy. By contrast, the computerised timetabling systems reviewed at other Universities are part of the official policy, (this was the reason for reviewing the particular systems).

The extent of the computerised timetabling system, and the amount of integration with other computerised information systems varies considerably across the systems reviewed. The Southwell system was designed to fill in the gaps in the Institute's Student Record System, and represents perhaps 20% of the total system. Of its five subsystems two are integrated, and three are not integrated.

At the University of Waterloo, the class timetabling is an intrinsic part of the Student Record System. However, the initial construction of the class timetable, and the whole process of the examination timetable, are not integrated.

The SUNY at Buffalo System is confined to examination scheduling and rooming, and the Berkeley System to class and examination scheduling and rooming. Berkeley operate a separate system for Advance Class Enrolment, with the timetable being fixed by supply rather than demand.

At Memorial, the system concentrates on the registration of students, particularly on the design of individual timetables for intake students. The timetable is demand driven. Boulder also has a comprehensive scheme for registration of students.

The policy of the universities mentioned above with regard to examination timetabling is generally to adhere to class timetables, thus five or six days is the normal duration. There is no attempt to assemble all students in a subject together at the same time for examination. At Nottingham, however, the policy is to schedule the examinations clash-free, which requires a longer period.

The Waikato System, although not computerised, is interesting for the ability to respond quickly to changing levels of student enrolment, and for the existence of very high levels of room occupancy.

There is no system (among those reviewed) which could be adopted at FIT without change, but there are several ideas which can be examined for possible improvements to the FIT system.

15.3. Further Work.

In the situation, for the examination scheduling problem, that the number of colors required to color the graph is greater than the maximum number allowed, or that the number of colors is specified, it may be necessary to improve the model or the algorithms. Possibly an optimal tree-search algorithm or an integer programming model may be used to determine the chromatic number. Alternatively, the heuristic method may be improved by re-ranking the vertices at each stage, or by interchange procedures.

Another approach is to search for a solution which is more satisfactory in a particular circumstance, perhaps by formulation as a set-covering problem, by determining the size of the largest clique, or by identifying the vertices contained in a clique of largest size in the graph.

The idea of clustering offers a method for the construction of an examination timetable for the whole institute, which has morning and afternoon examinations. Certain subjects form connections between large clusters and may be pre-assigned.

Class timetables may be constructed by computer. This method has been tried in the past by the author, with some success, but may be developed further. New methods of allocation, such as expert systems, simulated annealing and tabu search, may be tried.

The systems in use at other universities will be examined further, with a view to improving the system at Footscray Institute of Technology, and to making a further study in the area of computerised timetabling and room scheduling.

15.4. Conclusion.

Programs have been provided by which the examination and class timetables of the Faculty of Business at Footscray Institute of Technology may be worked out each semester. Systems have been implemented at Institute level for recording and reporting of examination and room scheduling data, and are said to have reduced the amount of work required and increased the accuracy of the operation.

A review of the literature on timetabling and related systems has shown some interesting mathematical models and practical solutions. Site visits to universities where some form of computerised timetabling is in operation have revealed some of the necessary conditions for successful implementation of the systems, and have indicated direction for further work in the area.

REFERENCES

ABRAMSON D.(1988), "A Language and Compiler System for Specifying Timetable Requirements", C.S.I.R.O., Melbourne.

ACKOFF R.J.(1967), "Management Misinformation Systems", Management Science, 14, 4, B147-B156.

ALMOND Mary (1965), "An Algorithm for Constructing University Timetables",Comp.J., 8, 331-340.

ALMOND Mary (1969), "A University Faculty Timetable", Comp.J., 12, 215-217.

ARANI Taghi, KARWAN Mark, LOTFI Vahid (1988), "A Lagrangian Relaxation Approach to solve the Second Phase of the Exam Scheduling Problem", E.J.O.R., 34, 372-383.

ARANI Taghi, LOTFI Vahid (1989), "A Three Phased Approach to Final Exam Scheduling", IIE Transactions, 21, 86-96.

AUBIN Jean, FERLAND Jacques A. (1989), "A Large Scale Timetabling Problem", Comp.Opns.Res.,16, 1, 67-77.

BALAS Egon, TOTH Paolo (1983), "Branch and Bound Methods for the Travelling Salesman Problem", Management Science Research Report No.MSRR 488, Carnegie-Mellon University.

BALAS Egon, YU Chang Sun (1984), "Finding the Maximum Clique in an Arbitrary Graph", Management Science Research Report No.MSRR 515, Carnegie-Mellon University.

BARRACLOUGH Elizabeth D. (1965), "The Application of a Digital Computer to the Construction of Timetables", Comp.J.,8, 136-146.

BARTHOLDI J.J., PLATZMAN L.K. (1982), "An O(NlogN) Planar TS Heuristic Based on Spacefilling Curves", Op.Res.Letters, 1, 121-125.

BAUM Sanford, CARLSON Robert (1979), "On Solutions that are Better Than Most", Omega, 7, No.3, 249-255.

BENDER Edward A., WILF Herbert S. (1985), "A Theoretical Analysis of Backtracking in the Graph Coloring Problem", Journal of Algorithms, 6, 275-282.

BERGE C. (1973), Graphs and Hypergraphs, North Holland, Amsterdam.

BILLIONNET Alain (1981), "An Upper Bound on the Size of the Largest Clique in a Graph", J.Graph Theory, 5, 165-169.

BOYD S.C., PULLEYBLANK W.R., CORNUEJOLS G. (1988), "TRAVEL:- Programs for Travelling Salesman Problems", University of Waterloo.

BRELAZ Daniel (1979), "New Methods to Color the Vertices of a Graph", Comm.A.C.M. 22, 251-256.

BRODER S. (1964), "Final Examination Scheduling", Comm.A.C.M., 7, 494-498.

BROOKS R.L. (1941), "On Coloring the Nodes of a Network", Proc. Cambridge Philosophical Society, 37, 194.

BROWN J.R. (1972), "Chromatic Scheduling and the Chromatic Number Problem", Man.Sci., 19, 456.

CAMPERS G., HENKES O., LECLERQ J.P. (1987), "Graph Coloring Heuristics: A Survey, Some New Propositions and Computational Experiences on Random and "Leighton's" Graphs", Operational Research '87, 917-932.

CAMPERS G., HENKES O., LECLERQ J-P. (1987b), "Sur les Methodes Exactes de Coloration de Graphes", Cahiers de C.E.R.O. 29, 19-30.

CARLSON Eric D. ed. (1977), Proceedings of a Conference on Decision Support Systems, Data Base, 8, 3.

CARTER M.W. (1978), "Examination Scheduling: Documentation", Internal Report, Data Processing Department, University of Waterloo.

CARTER Michael W. (1979), "Final Examination Scheduling for Correspondence Students", Research Report CORR 79-7, University of Waterloo.

CARTER Michael (1983), "A Decomposition Algorithm for Practical Timetabling Problems", Working Paper #83-06, Department of Industrial Engineering, University of Toronto.

CARTER Michael (1984), "A Practical Algorithm for Finding the Largest Clique in a Graph", Working Paper #84-08, Department of Industrial Engineering, University of Toronto.

CARTER Michael W. (1986), "A Survey of Practical Applications of Examination Timetabling Algorithms", Operations Research, 34, 2, 193-202.

CARTER Michael (1987), "Timetabling Literature", Internal Report University of Toronto.

CARTER Michael W. (1987), "A Lagrangian Relaxation Approach to the Classroom Assignment Problem", Working Paper #87-13, Department of Industrial Engineering, University of Toronto.

CATLIN Paul A. (1978), "Another Bound on the Chromatic Number of a Graph", Discrete Math., 24, 1-6.

CHAHAL N, dewerra D. (1989), "An Interactive System for Constructing Timetables on a PC", E.J.O.R., 40, 32-37.

CHIBA N., NISHIZEKI T., SAITO N.(1982), "An Approximation Algorithm for the Maximum Independent Set Problem in Planar Graphs", SIAM J.Comput., 11,4, 663-675.

CHRISTOFIDES N. (1971), "An Algorithm for the Chromatic Number of a Graph", Comp.J., 14, 38-39.

CHRISTOFIDES Nicos (1975), Graph Theory - An Algorithmic Approach, Academic Press, London.

CLEMENTSON A.T., ELPHICK C.H. (1983), "Approximate Coloring Algorithms for Composite Graphs", J.Op.Res.Soc.(UK), 34, 6, 503-509.

COLE A.J. (1964), "The Preparation of Examination Timetables using a Small-store Computer", Comp.J., 7, 117-121.

CORNEIL D.G., GRAHAM B. (1973), "An Algorithm for Determining the Chromatic Number of a Graph", SIAM J.Comput., 2, 4, 311-318.

CORNUEJOLS G., NEMHAUSER G. (1978), "Tight Bounds for Christofides Travelling Salesman Heuristic", Math.Prog., 14, 116-121.

DESROCHES,S., LAPORTE G., ROUSSEAU J.M. (1978), "HOREX - a Computer Program for the Construction of Examination Schedules", Infor., 16, 3, 294-298.

dBASE III (1984) Data Base Software, Ashton-Tate.

DEVINE Michael D., KUMIN Hillel J., LUTZ Raymond P. (1976), "A Time-sharing Interactive Program for Class Scheduling", Computers and Education, 1, 33-38.

DUTTON R.D., BRIGHAM R.C. (1981), "A New Graph Coloring Algorithm", Comp.J., 24, 85-86.

DYER J.S., MULVEY J.M. (1976), "An Integrated Optimisation/Information System for Academic Department Planning", Man.Sci., 22, 12, 1332-1341.

FERLAND Jaques, BABIN Gilbert, AUBIN Jean (1986), "A Course Scheduling System", Publication no. 590, Departmente d'informatique et de Recherche Operationnelle, Universitie de Montreal.

FISHER Marshall L. (1980), "Worst-case Analysis of Heuristic Algorithms", Man.Sci., 26, 1, 1-17.

FRIEZE A.M., GALBIATI G., MAFFIOLI F. (1982), "On the Worst-case Performance of some Algorithms for the Asymmetric Travelling Salesman Problem", Networks, 12, 23-39.

GELLER D.P. (1970), Problem 5713, Am.Math.Monthly, 77, 85.

GEOFFROY Dennis, SUMNER David (1978), "An Upper Bound on the Size of the Largest Clique in a Graph", J.Graph Theory, 2, 223-230.

GILLIAN R.J. (1970), "The Chromatic Number of a Graph", Msc Thesis, Imperial College, University of London.

GLASSEY C.R., MIZRACH M. (1986), "A Decision Support System for Assigning Classes to Rooms", Interfaces, 16, 5, 92-100.

GODIN Victor (1978), "Interactive Scheduling: Historical Survey and State of the Art", AIIE Trans, 10, 331-337.

GOSSELIN Karl (1986), "Allocation of Classrooms by Linear Programming", J.Opl.Res.Soc., 37, 6, 561-569.

GRAY Paul, KING William R., McLEAN Ephraim R., WATSON Hugh J. (1989), The Management of Information Systems, Dryden Press.

HELL P., KIRKPATRICK D.G. (1978), "Scheduling, Matching and Coloring", Colloquia Mathematica Societatis Janos Bolyai, 25. Algebraic Methods in Graph Theory, Szeged (Hungary).

HEMMERLING M.B. (1974), "The Generation by Computer of Timetables for South Australian Secondary Schools", PhD Thesis, Department of Computing Science, University of Adelaide.

HERNANDEZ J.P., PROTH J.M. (1982), "A Good Solution instead of an Optimal One", Interfaces, 12, 2, 37-42.

HIGHSMITH Jim (1987), "Structured Systems Planning", J.Info.Systems Management, Spring 1987, 23-33.

JOHNSTON H.C. (1976), "Cliques of a graph - Variations on the Bron-Kerbosch Algorithm", Intl.J.Comp.and Infor.Sciences, 5, 3 209-238.

JUNGINGER Werner (1986), "Timetabling in Germany - A Survey", Interfaces, 16, 4, 66-74.

KARP R.M (1975), "The Fast Approximate Solution of Hard Combinatorial Problems", Proc. 6th South Eastern Conference on Combinatorics, Graph Theory and Computing, Utilitas Mathematical, Winnipeg, 15-31.

KEEN P.G.W., MORTON Scott (1978), Decision Support Systems, An Organisational Perspective, Addison Wesley, Reading MA.

KEEN Peter G.W. (1980), "Decision Support Systems", Sloan Management Review, Spring 1980, 33-44.

KLEIN Michel, METHLIE Leif B. (1990), Expert Systems - A Decision Support Approach, Addison Wesley.

KNUTH D.E. (1973), The Art of Computer Programming, Vol.3, Sorting and Searching, Addison Wesley, Reading MA.

KORMAN S.M. (1977), "The Graph Coloring Problem", Internal Report, Imperial College, London.

LAPORTE G., DESROCHES S. (1982), "Examination Timetabling by Computer", H.E.C., Montreal.

LAPORTE G., DESROCHES S. (1984), "Examination Scheduling by Computer", Comp.and Op.Res., 11, 4, 351-360.

LAPORTE Gilbert, DESROCHES Sylvain (1986), "The Problem of Assigning Students to Course Sections in a Large Engineering School". Comp.and Op.Res., 13, 4, 387-394.

LARSON Jean A. (1979), "Some Graphs with Chromatic Number Three", J.Comb.Theory Ser.B., 26, 317-322.

LAWLER, LENSTRA, RINNOOY KAN, SHMOYS eds. (1985), The Travelling Salesman Problem, John Wiley and Sons.

LAWRIE N.L. (1969), "An Integer Linear Programming Model of a School Timetabling System", Comp.J., 12, 307-316.

LEIGHTON F.T. (1979), "A Graph Coloring Algorithm for Large Scheduling Problems", J.Res.N.B.S., 84, 6, 489-506.

LIN S., KERNIGHAN B.W. (1973), "An Effective Heuristic for the Travelling Salesman Problem", Op.Res., 21, 498-516.

LINDBLOM C.E. (1979), "Still Muddling, Not Yet Through", Public Administration Review, 39, 517-526.

LIONS John (1966), "Matrix Reduction Using the Hungarian Method for the Generation of School Timetables", Comm.A.C.M., 9, 5, 349-354.

LITKE J.D. (1984), "An Improved Solution to the Travelling Salesman Problem with Thousands of Nodes", Comm.A.C.M., 24, 1227-1236.

LITTLE John D.C.(1979), "Decision Support Systems for Marketing Managers", J.Marketing, 43, Summer 1979, 9-26.

LOTFI Vahid, CERVENY Robert (1989), "Implementing a Final Examination Scheduling Package", Internal Report, School of Management, State University of New York at Buffalo.

LOTFI Vahid, SARIN Sanjiv (1986), "A Graph Coloring Algorithm for Large Scale Scheduling Problems", Comp.and Op.Res., 13, 1, 27-32.

MARSTEN Roy E. (1974), "An Algorithm for Large Set Partitioning Problems", Man.Sci., 20, 5, 774-787.

MATULA D.W., MARBLE G., ISAACSON I.D. (1972), "Graph Coloring Algorithms", in Graph Theory and Computing READ R.C. ed., Academic Press, New York.

MAWSON Alen (1988), Time Chart, Software for Building and Printing Timetables, Amig Systems, Melbourne.

McCLURE R.H., WELLS C.E. (1984), "A Mathematical Programming Model for Faculty Course Assignments", Decision Sciences, 15, 3, 409-420.

McDIARMID Colin (1979), "Determining the Chromatic Number of a Graph", SIAM J.Comput., 8, 1, 1-14.

MEHTA Nirbhay K. (1981), "The Application of a Graph Coloring Method to an Examination Scheduling Problem", Interfaces, 11, 5, 57-64.

MEMORIAL UNIVERSITY OF NEWFOUNDLAND (1990), Undergraduate Registration Procedures and Class Schedule, Fall 1990.

MICROSOFT (1983), BASIC Compiler Users Guide, Microsoft Corporation.

MORTON Michael S Scott (1971), Management Decision Systems, Boston, Harvard University.

MYERS B.R., LIN R. (1972), "A Lower Bound on the Chromatic Number of a Graph", Networks, 1, 273.

NEUFELD G.A., TARTAR J. (1974), "Graph Coloring Conditions for the Existence of Solutions to the Timetabling Problem", Comm.A.C.M., 17, 8, 450-453.

NEUFELD G.A., TARTAR J. (1975), "Generalised Graph Colorations", SIAM J.App.Math., 29, 91-98.

NISHIZEKI T., CHIBA N. (1988), Planar Graphs: Theory and Algorithms, Elseviers, Amsterdam.

NORDHAUS E.A., GADDUM J.W. (1956), "On Complementary Graphs", Am.Math.Monthly, 63, 175.

OSTEEN R.E., TOU J.T. (1973), "A Clique-detection Algorithm based on Neighborhoods in Graphs", Intl.J.Comp.and Info.Sciences, 2, 4, 257-268.

PECK J.E.L., WILLIAMS M.R. (1966), Algorithm 286 - Examination Scheduling, Comm.A.C.M., 9, 433-434.

PEEMÖLLER J. (1986), "Numerical Experiences with Graph Coloring Algorithms", E.J.O.R., 24, 146-151.

REINGOLD Edward M. (1977), Combinatorial Algorithms: Theory and Practice, Prentice Hall.

ROMERO B.P. (1982), "Examination Scheduling in a Large Engineering School - a Computer Assisted Participative Procedure", Interfaces, 12, 17-24.

SABIN G.C.W., WINTER G.K. (1986), "The Impact of Automated Timetabling on Universities - A Case Study", J.Op.Res.Soc., 37, 7, 689-693.

SALAZAR A., OAKFORD R.V. (1974), "A Graph Formulation of a School Scheduling Algorithm", Comm.A.C.M., 17, 12, 686-698.

SALKIN Harvey M.,de KLUYVER C.A. (1975), "The Knapsack Problem - A Survey", Naval Res.Logistics Quart., 22, 127-144.

SCHMIDT G., STRÖHLEIN T. (1980), "Timetable Construction -an Annotated Bibliography", Comp.J. 23, 4, 307-316.

SCOTT Trevor B. (1976), "Graph Coloring with Preassignment and Unavailability Constraints", Ars Combinatoria, 2, 25-32.

SELKOW Stanley M. (1978), "New Bounds for the Clique Number of a Graph", Info.Proc.Lett., 7, 4, 173-174.

SILVER E.A., VIDAL V., deWERRA D. (1980), "A Tutorial on Heuristic Methods", E.J.O.R., 5, 153-162.

SOUTHWELL Mary (1988), "ROOMS - A Recording and Analysis System for Classroom Booking Data", Report 5, Footscray Institute of Technology.

SOUTHWELL Mary (1988L), "STUDREC - A Student Record System for a Faculty or Department", Report 4, Footscray Institute of Technology.

SRIMANI P.K., SINHA B.P., CHOUDHURY A.K. (1978), "A New Method to find the Chromatic Partition of a Symmetric Graph", Int.J.Systems Sci., 9, 12, 1425-1437.

SZEREKES G., WILF H.S. (1968), "An Inequality for the Chromatic Number of a Graph", J.Comb.Th., 4, 1-3.

TRIPATHY A. (1984), "School Timetabling - A Case in Large Binary Integer Programming", Man.Sci., 30, 12, 1473-1489.

TUTTE W.T. (1954), Solution of Advanced Problem No. 4526, Am.Math.Month. 61, 352.

UNIVERSITY OF CALIFORNIA AT BERKELEY (1987), Description of ACE Processing, Internal Report, Administrative Information Services.

UNIVERSITY OF CALIFORNIA AT BERKELEY (1990), Schedule of Classes, Fall 1990.

UNIVERSITY OF COLORADO AT BOULDER (1990), Catalog for 1990-1991, and Continuing Student Registration Instructions.

UNIVERSITY OF NOTTINGHAM (1985), Examinations Scheduling, private communication from the Examinations Officer.

UNIVERSITY OF WAIKATO (1990), Calendar 1990, and Timetable 1990.

VINCKE P. (1984), "Timetabling in Belgium - a Survey", Paper presented at TIMS XXVI Copenhagen.

WAGON Stanley (1980), "A Bound on the Chromatic Number of Graphs without Certain Induced Subgraphs", J.Comb.Theory Ser.B, 29, 345-346.

WANG (1974), "A Depth-first Tree Search Algorithm for the Chromatic Number", J.A.C.M., 21, 385.

WATSON S.R., BEUDE D.M. (1987), Decision Synthesis: the Principles and Practice of Decision Analysis, Cambridge, Cambridge University Press.

WELSH D.J.A., POWELL M.B. (1967), "An Upper Bound for the Chromatic Number of a Graph and its Application to Timetabling Problems", Comp.J., 10, 1, 85-86.

deWERRA D. (1975), "How to Color a Graph", Combinatorial Programming - Methods and Applications, 305-325.

deWERRA D. (1985), "An Introduction to Timetabling", Invited Review, E.J.O.R., 19, 151-162.

WHITE George M., CHAN Pak-Wah (1979), "Towards the Construction of Optimal Examination Schedules", Infor. 17, 3, 219-229.

WHITE George, HADDAD Michael (1983), "An Heuristic Method for Optimising Examination Schedules which have Day and Evening Courses", Comput.Educ., 7, 4, 235-238.

WIGDERSON A. (1983), "Improving the Performance Guarantee for Approximate Graph Coloring", J.A.C.M., 30, 4, 729-735.

WILLIAMS M.R. (1974), "Heuristic Procedures - If They Work, Leave Them Alone", Software Pract.Exp., 4, 237-240.

WOOD D.C. (1968), "A System for Computing University Examination Timetables", Comp.J., 11, 41-47.

WOOD D.C. (1969), "A Technique for Coloring a Graph Applicable to Large Scale Timetabling Problems", Comp.J., 12, 317-319.

ZYKOV A.A. (1952), "On Some Properties of Linear Complexes", Mat.Shornik 24/26, 163, American Math.Soc. Translation No 79.

FILES AND PROGRAMS IN THE EXAMINATION TIMETABLING SYSTEM

The Student Data File

Data Cleaning Program ——————> Subject Listing
HIT66B.BAS (optional)

Matrix Build Program ——————> Subject Listing
HIT77B.BAS Appendix B.

Graph Coloring Program ——————> Three alternative
HIT88B.BAS Colorings App.C.

Travelling Salesman Program ——————> Three alternative
HSP8.BAS Orders App.D.

Human Decision Phase ——————> Draft Examination
Timetable App.E.

Clash Check Program ——————> Adjustments
HIT99B.BAS (optional)

Appendix A. Files and Programs in the Examination Timetabling System.

Appendix B. Subjects found on File, Semester 2 1990.

Subject	Students	Subclash	Stuclash	Index.
HUS902B.DAT				
AE003	379	48	988	57
AE004	260	38	666	115
AE005	155	16	397	35
AE006	431	26	1092	15
AE049	15	0	0	50
AE101	2	1	1	140
AE125	21	1	4	91
AE126	130	42	331	59
AE151	36	41	82	19
AE171	37	1	32	90
AE186	58	18	167	108
AE221	32	26	79	27
AE225	12	14	31	22
AE231	15	15	25	48
AE232	28	28	52	23
AE234	46	44	123	20
AE250	28	35	78	83
AE258	72	46	186	41
AE270	13	14	38	134
AE271	22	20	64	131
AE272	29	22	73	56
AE285	9	8	24	130
AE324	18	20	44	54
AE325	12	15	24	39
AE341	1	1	1	94
AE342	3	6	7	110
AE343	10	19	23	44
AE345	16	20	32	84
AE346	1	2	2	82
AE347	17	24	39	3
AE348	29	30	64	70
AE364	22	0	0	31
AE365	5	6	9	100
AE370	29	33	77	40
AE446	14	23	41	107
AE447	46	27	118	26
AE486	23	29	57	80
AE526	18	3	18	1
AE542	22	2	21	139
AE543	11	2	10	66
AE547	10	1	9	67
AE586	4	0	0	127
AE620	11	3	22	122
AE622	1	2	2	32
AE625	1	1	1	147
AE626	18	3	18	2
AE627	14	5	25	33
AE629	15	3	22	34
AE700	2	0	0	13
AE941	17	1	12	68
AE942	1	2	2	124
AE943	47	1	1	5

AL001	159	19	346	87
AL002	495	39	1277	16
AL007	170	35	456	98
AL104	79	0	0	92
AL108	39	2	36	93
AL152	145	28	346	58
AL200	80	47	205	24
AL203	125	42	309	17
AL213	161	53	420	86
AL251	93	39	229	25
AL252	111	40	283	45
AL253	31	18	51	49
AL301	50	29	91	28
AL303	66	32	120	18
AL304	79	34	121	36
AL313	75	38	178	73
AL314	8	13	18	79
AL351	34	25	86	46
AL352	25	21	62	42
AL385	28	1	1	52
AL401	1	0	0	43
AL402	34	24	100	71
AL403	36	24	102	72
AL415	1	3	3	118
AL416	34	23	85	55
AL451	19	19	56	117
AL452	55	29	163	120
AL509	12	4	44	141
AL550	12	4	44	142
AL553	34	2	32	77
AL555	1	1	1	146
AL556	11	7	13	14
AL563	2	1	2	121
AL905	60	1	22	38
AL951	25	3	26	51

FB301	20	0	0	113
FB305	67	0	0	76
FB306	35	5	5	112
FB309	52	5	6	133
FB310	28	0	0	116
FB702	14	5	70	7
FB703	14	5	70	8
FB707	14	5	70	9
FB708	14	5	70	10
FB710	14	5	70	11
FB711	14	5	70	12
FB721	14	0	0	128
FB723	3	0	0	21
FB730	5	3	4	99
FB731	6	2	5	137
FB732	5	2	5	138
FB737	1	0	0	62
FB739	9	2	7	4
FB744	6	1	6	64
HA008	162	28	432	74
HA151	19	3	3	123
HA173	1	1	1	129
HA175	203	0	0	135
HA177	5	0	0	126
HA181	72	21	206	6
HA182	59	22	173	132
HA190	28	30	74	109
HA220	34	43	107	53
HA231	120	46	190	95
HA271	146	46	393	97
HA282	65	27	191	119
HA283	33	22	94	111
HA290	27	27	77	103
HA295	35	27	98	75
HA297	80	30	190	47
HA420	15	17	46	60
HA422	53	33	144	89
HA423	1	3	3	136
HA431	20	17	56	102
HA432	61	38	131	88
HA433	12	12	35	114
HA434	40	34	100	96
HA435	42	28	116	30
HA441	28	29	88	106
HA442	10	11	28	104
HA473	21	24	54	101
HA476	10	15	26	81
HA478	16	18	42	61
HA484	24	19	75	105
HA492	16	16	25	85
HA493	38	28	110	78
HA562	16	2	13	37
HA564	6	1	6	65
HA574	11	4	41	143
HA578	10	4	39	144
HA584	20	1	17	63
HA589	11	4	42	145
HA591	17	1	17	125
HA700	1	0	0	29
HA930	17	4	17	69

Appendix C. The Output from the Graph Coloring Program.

(Three Alternatives).

1. FIFO Order.

```
     HTEM902B.DAT
 1  20  418   418 AE049 AE151 AE221 AE325 AE347 AE364 AE526 AE622 AE700 AE943
AL352 AL401 AL905 FB702 FB723 FB739 HA181 HA420 HA562 HA700
 2  11  540  540 AE006
AE234 AE341 AE346 AE626 AE629 AE942 AL415 AL556 FB703 FB737
 3  5  540  540
AE225 AE627 AL002 FB707 FB730
 4  17  540  540 AE005 AE125 AE324 AE365 AE543 AE941 AL203
AL385 AL553 AL563 AL951 FB708 FB744 HA423 HA435 HA478 HA584
 5  16  540  540 AE171 AE232 AE342 AE547 AE586 AL104
AL200 AL303 AL416 FB305 FB710 HA177 HA297 HA442 HA564 HA930
 6  13  540  540 AE272 AL001 AL108
AL251 AL301 FB301 FB306 FB310 FB711 HA295 HA433 HA476 HA492
 7  9  540  540
AE003 AE370 AE447 AE486 AE620 AL253 AL451 AL555 HA173
 8  14  537  537 AE101 AE258 AE542 AE625
AL304 AL509 FB721 FB731 HA151 HA175 HA190 HA484 HA493 HA591
 9  12  398  398 AE186 AE231
AE270 AE285 AE343 AL252 AL550 FB309 FB732 HA182 HA220 HA431
 10  8  429  429
AE271 AE345 AL152 AL351 AL402 HA271 HA473 HA574
 11  5  249  249
AE126 AE348 HA290 HA422 HA578
 12  5  302  302
AL403 HA008 HA282 HA441 HA589
 13  3  191  191
AL313 AL452 HA432
 14  3  69  69
AE250 AL314 HA283
 15  2  201  201
AL213 HA434
 16  2  134  134
AE446 HA231
 17  1  170  170
AL007
 18  1  260  260
AE004
  0  0  0  0

  0  0  0  0
```

2. Decreasing Subclash Order.

```
 1  16  540  540 AE231 AE272 AE285 AE342 AE346 AE622
AE942 AL001 AL213 AL314 AL415 FB306 FB309 HA423 HA434 HA484
 2  6  540  540
AE003 AE370 AE625 AL203 AL555 FB730
 3  19  540  540 AE005 AE101 AE151 AE324 AE542 AE543 AE627 AL200 AL352
AL556 FB711 FB732 HA173 HA297 HA420 HA564 HA589 HA591 HA930
 4  16  540  540 AE126 AE234 AE271 AE341 AE347 AE365
AE526 AE620 AL108 AL553 AL951 FB710 FB731 HA271 HA478 HA578
 5  17  540  540 AE125 AE221 AE225 AE348 AE586 AE626 AE629
AL152 AL563 FB708 FB739 HA231 HA432 HA442 HA473 HA562 HA574
 6  12  540  540 AE004 AE171
AE258 AL304 AL401 AL550 FB707 FB737 FB744 HA151 HA493 HA700
 7  20  526  526 AE049 AE270 AE343 AE364 AE547 AE941 AL252 AL253 AL385 AL509
AL905 FB301 FB310 FB703 FB721 HA177 HA220 HA290 HA295 HA584
 8  10  495  495
AE700 AE943 AL007 AL251 AL301 FB702 FB723 HA181 HA441 HA492
 9  3  533  533
AE250 AL002 HA476
 10  7  507  507
AE186 AE486 AL313 AL416 AL452 HA175 HA182
 11  6  313  313
AL104 AL303 FB305 HA190 HA422 HA431
 12  7  329  329
AE232 AE325 AE345 AL403 HA008 HA233 HA435
 13  4  137  137
AE446 AE447 HA282 HA433
 14  3  499  499
AE006 AL351 AL402
 15  1  19  19
AL451
 0  0  0  0

 0  0  0  0

 0  0  0  0

 0  0  0  0

 0  0  0  0
```

3. Decreasing Stuclash Order.

```
 1  3  540  540
AE342 AL002 HA435
 2  4  540  540
AE006 AE250 HA297 HA423
 3  5  540  540
AE003 AE370 AL203 AL415 HA564
 4  11  540  540 AE004
AE258 AE365 AE627 AE942 AL108 AL301 AL550 FB711 HA422 HA431
 5  14  540  540 AE171 AE270 AE325 AE343
AL007 AL252 AL253 AL509 AL553 AL951 FB710 FB739 HA484 HA493
 6  12  540  540 AE231 AE272
AE542 AE620 AL313 AL416 FB708 FB732 HA008 HA271 HA492 HA589
 7  15  540  540 AE005 AE285 AE543 AE625 AE626
AE629 AL213 AL403 AL563 AL905 FB707 HA290 HA442 HA476 HA574
 8  17  540  540 AE234 AE347 AE526 AE547 AE622 AL001 AL251
AL352 AL556 FB703 FB744 HA282 HA420 HA562 HA578 HA591 HA930
 9  19  540  540 AE101 AE186 AE341 AE346 AE586 AE941 AL152 AL304 AL401
AL555 FB306 FB702 FB731 FB737 HA173 HA181 HA432 HA473 HA584
 10  18  539  539 AE049 AE125 AE126 AE271 AE486 AL104 AL385 AL451
AL452 FB301 FB309 FB310 FB721 FB723 FB730 HA151 HA177 HA700
 11  8  521  521
AE232 AE700 AE943 AL200 AL303 FB305 HA175 HA441
 12  7  258  258
AE324 AE364 AL314 AL351 HA231 HA434 HA478
 13  4  151  151
AE447 HA182 HA220 HA433
 14  6  162  162
AE221 AE225 AE345 AL402 HA283 HA295
 15  3  78  78
AE151 AE446 HA190
 16  1  29  29
AE348
 0  0  0  0

 0  0  0  0

 0  0  0  0

 0  0  0  0
```

Appendix D. The Output from the Travelling Salesman Program. (Three Alternatives).

```
 HEMP902.DAT
SUBJECTS TAKEN IN FIFO ORDER
 ROUTE    1  17   5  14   2  15   3  16  13  18   8   7  11   4  12   9   6  10    TOTAL   374

 ROUTE    2  14   3  15  17   1  18   8  16  13   9  12   5   7  11   4  10   6    TOTAL   373

 ROUTE    9  14   2  15   3  16  13  17   1  18   8   7  11   4  12   5  10   6    TOTAL   356

IN DECREASING SUBCLASH ORDER
 ROUTE    1  15   2  13  14  11   9   7  12   5  10   8   6   3   4    TOTAL   584

 ROUTE    2  15   7   9  11  13  14   3  10   6  12   5   1   5   4    TOTAL   580

 ROUTE    3  15   2  13  14  11   9   1  12   7  10   8   5   6   4    TOTAL   579

 ROUTE    4  15   2  13  14  11   9   1  12   7  10   8   5   3   6    TOTAL   537

IN DEC STUCLASH ORDER
 ROUTE    1  16   9  15   2  11  13  14  12  10   3   4   7   6   5   8    TOTAL   658

 ROUTE    4  16   1  11   2  15  13  14   9  10   3  12   7   6   5   8    TOTAL   561

D:\EXAM>
```

APPENDIX E

DRAFT EXAMINATION TIMETABLE (UNDERGRADUATE) 1990 SEM 2

Please note that this draft timetable is attached in the handwritten form in which it is normally issued within the Faculty of Business. The Institute Provisional Timetable is issued in computer-printed form, and the Final Timetable is issued in type-set form.

DRAFT EXAM. TIMETABLE - FOB-UNDERGRADUATE 1990 Sem 2.

Day + Date	Total	Subjects.
Mon 29 Oct.	47	HA190^{28} AL451$^{\text{Open } 19}$ (Setting Up Day)
Tues 30 Oct	198	AL104$^{2\ 79}$ AL108$^{2\ 39}$ HA297^{80} (Limited seating).
Wed 31 Oct.	523	AE250^{28} AL002^{495}
Thu 1 Nov	406	AE272^{29} AE283^{9} AL001^{159} AL213^{161} AL314$^{2 \text{ Open } 8}$ HA434^{40}
Fri 2 Nov	477	AE004^{260} AE258^{72} AL304^{79} AL385^{23} HA493^{35} (MORNING D0104)
Mon 5 Nov	340	~~AE270~~^{13} AE364^{22} AE941$^{1.5\ 17}$ AE171^{37} AL252^{111} AL253^{31} AL905$^{2\ 60}$ HA290^{27} HA295$^{3C\ 35}$
Tues 6 Nov	—	Cup Day Public Holiday.
Wed 7 Nov	533	AE003^{379} AE370^{29} AL203$^{O\ 125}$
Thu 8 Nov	292	AE446$^{2\ 14}$ AE447^{46} HA282^{65} HA433^{12} AE943$^{1.5\ 47}$ AE186^{58}
Fri 9 Nov	442	~~AE943~~$^{1.5\ 47}$ AE270^{13} AL007$^{\text{Open } 170}$ AL251$^{\text{Open } 93}$ AL301^{50} HA181^{72} HA441^{28} HA492^{16}
Mon 12 Nov	532	AE006^{431} AL351$^{\text{Open } 34}$ AL402^{34}
Tue 13 Nov	191	AL303$^{\text{Open } 66}$ HA422^{53} HA431^{20}.
Wed 14 Nov	413	AE125^{21} AE221^{32} AE225^{12} AL152$^{\text{Open } 145}$ HA231^{120} HA432^{61}
Thu 15 Nov	354	AE005^{155} AE324^{18} AL200^{80} AL352$^{\text{Open } 25}$ ~~HA420~~^{15} ~~HA430~~^{17}
Fri 16 Nov.	355	AE232^{28} AE325^{12} ~~AE345~~^{16} AL403^{36} HA008^{162} HA283^{33} HA438^{42}
Mon 19 Nov	361	AE126^{130} AE234^{46} AE271^{22} AE347^{17} HA271^{146}
Tu 20 Nov.	223	~~AE186~~^{58} ~~AE486~~^{23} AL313^{75} AL416^{34} AL452$^{\text{Open } 55}$ HA182^{59}
Wed 21 Nov	34	HA220$^{2\ 34}$
Thu 22 Nov	-	Available for alterations.
Fri 23 Nov	-	

Postgraduate is a separate timetable.

Subjects not listed are not being examined.

All exams start 1 pm. Duration 15 mins reading and 3 hours writing.

2 = 2 hours O = Open Book. Superscript = number of students.

Timetable will be sent to Student Administration 14/9/90.

Mary Southwell 12/9/90. E2.

VICTORIA UNIVERSITY OF TECHNOLOGY

Footscray Institute of Technology
Ballarat Road,
Footscray.
P.O. Box 64
Footscray, Victoria,
Australia 3011
Telex: AA 36596
Fax: (03) 689 4069
Telephone: (03) 688 4200

Office of the Registrar
RC:mci
11th June 1991.

MEMORANDUM

TO: *Dr Peter Cerone, Senior Lecturer, Department of Mathematics, Computing and Operations Research*

FROM: *Michael J Halls, Registrar*

SUBJECT: *DEVELOPMENTAL WORK - MARY SOUTHWELL*

I write concerning the submission for assessment for Master of Applied Science by Mary Southwell. During recent years, Mary has been an active participant in the development of automated computer based information management systems at the Institute. This has included formative work on timetabling and examination scheduling databases, which has provided a substantial part of the basis from which these significant tasks are currently undertaken. In providing assistance to the FIT administration, Mary has applied her technical competency together with her organisational abilities to the production of practical solutions to immediate problems she perceived at the Institute.

As I have indicated, the results of Mary's contribution to the development of scheduling systems has been substantial, and has provided the basis for improvements in the efficiency of administrative practices in these areas within the Division of the Registrar at FIT over a significant period. The fact that a number of systems currently in use are based on the work provided by Mary is ample testament to the practical value of her work.

In conclusion I would add that I consider that the Institute has benefited both directly and indirectly from the contribution Mary has made through her work on the design and implementation of database applications, in that the heightened awareness and interest that her work generated has added to the more immediate improvements in work efficiency which resulted from her involvement.

[redacted]

Michael J Halls