

This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (<u>https://dspace.lboro.ac.uk/</u>) under the following Creative Commons Licence conditions.

COMMONS DEED
Attribution-NonCommercial-NoDerivs 2.5
You are free:
<ul> <li>to copy, distribute, display, and perform the work</li> </ul>
Under the following conditions:
<b>Attribution</b> . You must attribute the work in the manner specified by the author or licensor.
Noncommercial. You may not use this work for commercial purposes.
No Derivative Works. You may not alter, transform, or build upon this work.
<ul> <li>For any reuse or distribution, you must make clear to others the license terms of this work</li> </ul>
<ul> <li>Any of these conditions can be waived if you get permission from the copyright holder.</li> </ul>
Your fair use and other rights are in no way affected by the above.
This is a human-readable summary of the Legal Code (the full license).
<u>Disclaimer</u> 曰

For the full text of this licence, please go to: <u>http://creativecommons.org/licenses/by-nc-nd/2.5/</u>

UNI	LOUGHBOROUGH ERSITY OF TECHNOLO LIBRARY	GY
AUTHOR/FILI	IG TITLE	
	MAIL, Z.H.	
ACCESSION/C	OPY NO.	
	040129424	
VOL. NO.	CLASS MARK	
	ARCHIVES	
	CORY	
FO	REFERENCE ONLY	
101		¥
· · · ·		

· · · · ·	0401294242	· .		

· ·

# ASPECTS OF COMPUTERISED TIMETABLING

by

## Zuhaimy H. Ismail B.Sc. (Hons), M.Sc., PGCE

## A Doctoral Thesis Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy

December 1994

© by Z.H.Ismail, 1994

Comments and an
Loughbarnah
of Technology University
Construction of the second sec
JA96
Class
Acc.
No. 0to129421

.

Dedicated to my wife, Aini, my children, Hanis, Hanan, Attiyaa and Aliyaa my mum, Hajah Habsah Abdul Rahman and my father, Haji Ismail Imam Hassan

## ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and thanks to Mr David G. Johnson who has acted as my supervisor and who made himself available constantly for consultation. His help, patience, comments and suggestions on various topics have been greatly appreciated in the preparation of this thesis.

I would also like to thank my research panel under the directorship of Dr John M. Wilson and Dr David Coates. They made significant suggestions and criticisms during the progress of the research.

I would like to thank Professor Datuk Ridzuan Salleh, Professor Haji Muhammad Noor Salleh, Professor Hasanni and Dr Sallehi for their continuous support and encouragement for me to continue this work regardless of all the uncertainties. I would also like to acknowledge The Government of Malaysia and Universiti Teknologi Malaysia for their financial support during the period of this research. This has made it possible for me to pursue my PhD studies at Loughborough University.

Finally, I would like to thank my wife, Wan Aini Wan Ibrahim for her continuous and endless support and to both my parents for their support and prayer throughout my education. I would not have reached this stage without their patience, sacrifices and encouragement. May Allah bless them both and put them among the people of taqwa. To my children, Hanis, Hanan, Attiyaa and Aliyaa for their patience in putting up with my studies.

i

## ABSTRACT

This research considers the problem of constructing high school timetables using a computer. In the majority of high schools, termly or yearly timetables are still being produced manually. Constructing a timetable is a hard and time consuming task which is carried out repeatedly thus a computer program for assisting with this problem would be of great value. This study is in three parts. First, an overall analysis of the problem is undertaken to provide background knowledge and to identify basic principles in the construction of a school timetable. The characteristics of timetabling problems are identified and the necessary data for the construction of a timetable is identified. The first part ends with the production of a heuristic model for generating an initial solution that satisfies all the hard constraints embodied in the curriculum requirements.

The second stage of the research is devoted to designing a heuristic model for solving a timetable problem with hard and medium constraints. These include constraints like the various numbers of common periods, double periods and reducing the repeated allocation of a subject within any day. The approaches taken are based on two recently developed techniques, namely tabu search and simulated annealing. Both of these are used and comparisons of their efficiency are provided. The comparison is based on the percentage fulfilment of the hard and medium requirements.

The third part is devoted to one of the most difficult areas in timetable construction, that is the softer requirements which are specific to particular schools and whose satisfaction is not seen as essential. This section describes the development of an expert system based on heuristic production rules to satisfy a range of soft requirements. The soft requirements are studied and recorded as

ii

rules and a heuristic solution is produced for each of the general requirements. Different levels of rule are developed, from which the best possible solution to a particular timetable problem is expertly produced.

Finally, possible extensions of the proposed method and its application to other types of the timetabling problem are discussed.

# CONTENT

# PAGE

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
INTRODUCTION	1
1.1 Introduction	1
1.2 Educational Timetabling	
1.3 Some Terminology	5
1.3.1 Common Definitions	б
1.3.2 School Timetabling	7
1.4 What makes timetabling difficult ?	9
1.4.1 NP-Completeness	9
1.4.2 Constraints	10
1.5 Organisation of The Thesis	11
REVIEW OF THE LITERATURE	13
2.1 Introduction	13
2.2 The Role of the Computer in Timetabling	13 August 13
2.3 Early Computerised Timetabling in Schools	14
2.3.1 The Teddington System	16
2.3.2 The Oxford System Group	17
2.3.3 The Nor-Data System	17
2.3.4 The Strathclyde Program	18
2.3.5 The SPL System	- 19
2.3.6 The SIMS Approach	19

2.3.7 Other Systems	20
2.4 Recent Techniques for Solving Timetabling Problems	21
2.4.1 Decision Support Systems	22
2.4.2 Spreadsheet Approach	22
2.4.3 Graph Colouring	23
2.4.4 Expert Systems	24
2.4.5 Logic Programming	25
2.4.6 Knowledge-Based Approaches	27
2.4.7 Heuristic Approaches	29
2.4.7.1 Heuristics based on Tabu Search	29
2.4.7.2 Heuristics based on Simulated Annealing	31
2.4.7.3 Heuristics based on Genetic Algorithms	34
2.4.7.4 Heuristics based on Neural Networks	35
2.4.8 Database Approaches	36
2.5 Summary	37
ACKGROUND WORK	38
3.1 Introduction	38
3.2 Research Aim	38
3.3 Nature of The Problem	39
3.4 The Timetabling Process	41
3.4.1 Curriculum Planning	41
3.4.2 Staff Deployment Analysis.	41
3.4.3 Timetable Feasibility Test.	41
3.4.4 Assignment Task.	. 42
3.5 Solution Method	42
2 E 1 Cinculated American and Take Connels	11

3.5.3 Rule-Based Expert Systems	47
3.6 The Proposed System	48
3.6.1 Input Sub-system	48
3.6.1.1 Input Module	49
3.6.1.2 The Modifying Module	50
3.6.2 Solution Sub-system	50
3.6.2.1 Initial Solution Module	50
3.6.2.2 Intermediate Solution Module	51
3.6.3 Rule-Based Expert Sub-system	51
3.6.3.1 Rule Building Module	51
3.6.3.2 Rule-Based Solution Module	52
3.6.3.3 Rule Evaluation Module	52
3.6.4 Output Sub-system	52
3.6.4.1 Manual Operation Module	52
3.6.4.2 Graphic Display Module	53
3.7 System Testing	53
3.8 Summary	53
SOLUTION APPROACH BASED ON TABU SEARCH FOR SOLVING	
TIMETABLING PROBLEMS	55
4.1 Introduction	55
4.2 Solution Development	55
4.2.1 Basic Data Input	58.
4.2.2 Starting Solution	59
4.2.3 Producing a Feasible Timetable	60
4.2.4 Deviance matrix	61
4.3 Satisfying Medium Requirements	64
4.3.1 Satisfying Spread	64

4.3.2 Satisfying Double Periods	65
4.4 Results and Discussion	69
4.5 Summary	71
FURTHER REFINEMENT OF THE TIMETABLE USING SIMU	LATED
ANNEALING	74
5.1 Introduction	74
5.2 The Timetabling Annealing Scheme	74
5.2.1 Definitions	74
5.2.2 The objective function	75
5.2.3 Initial allocation	78
5.3 Adaptation of simulated annealing	78
5.3.1 Cooling Schedule	79
5.3.2 Initial Values of the Control Parameter	79
5.3.3 Decrement of the Control Parameter	80
5.3.4 Termination of the Run	81
5.3.5 Condition for Accepting A Move	81
5.4 Satisfying Spread and Double Periods	82
5.5 Results and Discussion	85
RULE-BASED EXPERT SYSTEM FOR SOLVING TIMETABLING PROBLEMS	87
6.1 Introduction	20 1. Sector 87
6.2 Expert System - A Definition	88
6.3 Characteristics of Expert System	89
6.4 Knowledge Representation in Expert Systems Technology	89

6.4.1 Production Rules	. 90
6.4.2 Advantages of a Rule-Based Approach	92
6.5 Rule-Based System Development For Timetabling	93
6.5.1 Type of Rules	93
6.5.1.1 Rule_One	95
6.5.1.2 Rule_Two	95
6.5.1.3 Rule_Three	96
6.5.1.4 Rule_Four	97
6.5.1.5 Rule_Five	98
6.5.2 Combinations of Soft Requirements	98
6.5.3 Combinations of Three or More Soft Requirements	100
6.6 Example and Discussions	101
6.7 Summary	103
AUTOMATIC COMPUTERISED TIMETABLING SYSTEM IMPLEMENTATION AND CASE STUDIES	્ર ભરજ
7.1 Introduction	105
7.2 The System Development	105
7.2.1 Users and their characteristics	108
7.2.2 Identify user task.	108
7.2.3 Continuous User Involvement	109
7.3 Introduction to ACT System	109
7.4 ACT System Architectures	110
7.4.1 Input Module	111
7.4.1.1 Curriculum Input sub-module	111
7.4.1.2 Expert Rule Input sub-module	112
7.4.2 Modification Module	112
7.4.2.1 Requirement Modification Sub-module	112

;

ţ

·	
7.4.2.2 Output Modification sub-module	113
7.4.3 Processes Module	115
7.4.3.1 Heuristic Processes Sub-module	115
7.4.3.2 Rule based Processes Sub-module	116
7.4.4 Output Module	117
7.4.4.1 Other Options	118
7.5 Case Studies	118
7.5.1 Timetabling at Limehurst High School	119
7.5.2 Timetabling at Market Bosworth High School	125
7.5.3 Timetabling at Stonehill High School	130
7.6 Summary	134
SUMMARY AND CONCLUSIONS	136
8.1 Introduction	136
8.2 Overview of The Research	136
8.3 Review and Evaluation of Case Studies	139
8.4 Discussion and Evaluation of Heuristic Methods Used	139
8.5 Discussion and Evaluation of the ACT System	142
8.5 Future Work	143
8.6 Conclusions	144
BIBLIOGRAPHY	146
APPENDIX A	156
APPENDIX B	159
APPENDIX C	161
APPENDIX D	163

Introduction

## **CHAPTER 1**

## INTRODUCTION

## 1.1 Introduction

Timetabling is a common activity in many organisations where groups of people must be scheduled for a number of meetings during a given time period with a given set of resources such as meeting rooms. Constructing a timetable is usually accompanied by a set of constraints which stipulate such things as the amount of time allowed per meeting, the required attendance at the meetings and the mix of resources. Some examples of timetabling problems are the scheduling of public transport facilities such as airplanes, trains and buses, meetings of managers within large organisation, timetabling in schools and universities, the scheduling of hospital facilities, scheduling the meetings of large societies and scheduling machines in factories to different groups of users. All of situations, timetabling in educational establishments such as schools and universities is undoubtedly the most common situation where timetabling activities take place.

The construction of a timetable in the majority of secondary schools is a long and complex process and, because it is based on the small units such as a single class, the individual teacher and the individual subject, it demands a multitude of decisions. As computer technology has advanced in terms of processing speed and storage, the use of computer technology could be of great help in solving this complex problem. It is widely recognised that automatic timetabling has been a difficult problem to solve (Falcao, 1990). Many approaches have been followed ranging from the use of formal operational research techniques such as graph

## Introduction

colouring (Almond, 1966) to solve course, teacher and classroom conflicts, to less formal techniques relying exclusively on heuristic methods (Cravo and Martin, 1987). Using formal approaches such as graph colouring is useful for understanding and defining the problem, although when applied to real circumstances, it may perform rather poorly. This is the case with many other algorithms where the number of variables and constraints to be considered grows exponentially in relation to the size of the school as defined by the numbers of classes, teachers and classrooms. Even when using heuristics to control the combinatorial explosion, this approach remains rather inefficient in terms of computational time. Nevertheless the use of formal mathematical methods such as graph colouring, has proved to be useful in solving certain parts of the problem.

This project is an outcome of an initial study on the construction of school timetables in several high schools in Leicestershire. Based on information gathered from timetablers in six different schools in the county, it appears that construction of an effective timetable for a school is the most important activity before a school year begins. Most schools in the county still construct their timetable manually and it can involve a considerable amount of time. Even if the task is undertaken by the most experienced timetabler, the time taken could easily be 100 hours or more for a medium-sized school. As the size of schools increases, timetable construction has become even more complex and often calls for an improvement in school management and administration. Since manually generated timetables are in use almost everywhere at the present time, there appears to be considerable scope for the efficient use of existing computer technology as an alternative approach.

With the introduction of the Education Reform Act which requires the school to manage its own finances, several management and administration computer programmes have been developed. Timetabling packages are sometimes included as part of these programmes, but they are invariably timetable management and

#### Introduction

printing systems which link in with the student database. Such packages do not usually have the capability of actually devising a timetable, a task which must first be performed manually, or largely so.

The main area of this research is to investigate simple heuristics based on the most recent optimization techniques and at the same time to produce a program that can be used interactively by the timetablers in schools. It is unlikely that any system could fully generate a complete timetable without any adjustment manually (no computer program can fully satisfy all requirements (de Werra, 1986)). The interactive element is therefore important in allowing the timetabler freedom to fine-tune the timetable to meet a range of local requirements. This work is expected to reduce significantly the large number of man-hours required to prepare and modify the timetable.

In this work, the construction of an automatic computerised timetable for high schools is based on a hybrid of three approaches; namely tabu search, simulated annealing and the hybrid of tabu Search and simulated annealing with a rule-based expert system approach. The very different types of constraints in timetabling, depending mainly on the type and size of school and the varying needs of different schools, will mean that the approach developed is not to attempt to find any kind of ideal solution, but merely an acceptable one.

## 1.2 Educational Timetabling

Timetabling problems are sometimes known as *resource allocation* problems and in some literature are also described as *scheduling* problems. Different definitions of timetabling problems have been provided depending on the problem and how it is being solved. One definition of a timetabling problem is as follows:

Introduction

"...the scheduling of a certain number of meetings, which are to be attended by a specific group of students and a teacher (or teachers), over a definite period of time, requiring certain resources (e.g. rooms, teaching aids etc.) in conformity with the availability of resources and fulfilling certain other requirements."

[Tripathy, 1984].

The process of constructing a timetable would not be difficult if no constraints were attached to it but this is unlikely to be the case in practice. Any timetable construction is usually subject to a set of constraints which are often divided into several categories such as the *hard*, the *medium*, and the *soft* constraints (Eiselt and Laporte,1986; Hertz A., 1992). The hard constraints are constraints that must be satisfied and a timetable will not be considered acceptable if they are violated, while the medium or soft constraints may be violated if absolutely necessary. A timetabler may have to repeat the process again and again until a workable timetable is achieved which satisfies the hard constraints and most, if not all, medium and soft requirements.

Educational timetabling problems differ widely, but they have several factors in common. First, they are very difficult problems to solve exactly or optimally. Most schools solve their timetabling problems by using human experts who have gained much experience and intuition into their own specific situation and (are) so able to produce a more effective timetable than some suboptimal automatic procedures. Often the constraints on the problem change so frequently that any feasible solution is sufficient. Second, in solving these problems, quite often heuristics and rules of thumb are used which are usually very difficult to formalise mathematically. Third, even when an analytic formulation of the problem is attempted, many intricate constraints are not formalised and only very simple constraints are used. In cases where a solution is not found, the expert (timetabler) often has to use judgement in relaxing some of the constraints, something which is difficult within the framework

## Introduction

of an analytical approach. Finally, timetabling problems are known to be virtually impossible to solve by exhaustive search. These difficulties are present even in a relatively small size problem such as examination scheduling in a college.

The construction of a timetable requires two stages. First, the curriculum planning stage in which the head teacher sets out the broad requirements that the timetable should satisfy. This is usually the result of discussion with heads of department to determine factors such as organisation of option groups, number of periods of each subject to be taught to each class or year group, allocating of staff, etc.. This stage usually does not involve the real timetabling process but more the administrative aspects. The second stage begins when a firm set of requirements or constraints has been agreed, and the timetabler tries to see whether a feasible timetable can be created from the requirements defined in the curriculum. This stage involves the detailed checking of the feasibility of the requirements specified in stage one, and is a tedious and time consuming process when done by hand.

## **1.3** Some Terminology

It is important at the outset to define the standard timetabling terminology used in this thesis. It must also be emphasised that this relates to school based timetabling where each year band follows a defined curriculum.

- Timetables at educational establishments are usually prepared on a weekly basis with a specified number of days per week. The same timetable will be used for either a term or a full school year.
- A subject is a body of knowledge taught to a group of students during a term or school year (e.g. Mathematics, Science, Games)
- A class is a group of pupils who are taught an identical set of subjects together.

### Introduction

### Chapter 1

- A period is a set time interval during a day where a lesson is taking place. The time is usually fixed to a consistent time span (with it's starting and ending time defined) throughout the week and there is therefore a fixed number of periods allowed in a day or week. If a topic requires more than a single period, two consecutive periods may be combined to make it into a double period. Some lessons must be taught at the same time to the whole school and a special period is assigned for that. This is referred to as a common period.
- A year group consists of several classes usually based on the age of pupils. These year groups are often sub-divided by the timetabler into a number of (usually two or three) year bands.
- A lesson is the basic teaching unit which consists of a subject taught during a period.(e.g. One period of English, Double period of Design).
- The frequency of a subject for each year band or class is the number of lessons in the subject taught within a week.
- The curriculum for each year band or class is the list of subjects to be taught to this group with their associated frequencies. This information is often stored in the form of a matrix.
- The Curriculum Matrix is a two dimensional array containing the number of teaching periods for all the subjects taught to each year group.
- A timeslot refers to a specific period in a day or a week where a class is taught a particular subject. Sometimes the word slot is also used.

## **1.3.1** Common Definitions

A number of definitions relating to timetable construction will be used in this thesis. Each chapter will require some special definitions for the terms used. Below is, a list of common definitions that will be used throughout the thesis.

#### Introduction

## Chapter 1

- the number of days per week ( weekly cycle = D)
- the number of periods per week (total periods = T)
- the number of year bands (total year bands = C)
- the number of subjects (available subjects = S)
- the number of periods per day (daily periods = P)
- the curriculum requirement for year band i and subject j (curriculum matrix  $A = a_{ij}$ )

A school timetable may be represented in a matrix form R, with rows representing the year bands and columns the period numbers. The size of this matrix depends on the number of year bands (C) and the number of weekly periods (T). Each element in the matrix represents the subject taught to a particular year band in  $\alpha$ certain period.

## **1.3.2** School Timetabling

To illustrate a typical school timetabling problem, consider the case of local high schools taking pupils from 11-14 years in three separate year groups (years 7, 8 and 9). Each year group is divided into a number of bands (typically 2 or 3) and within each band there are a number of classes (typically 3 or 4). The whole school therefore comprises between 400 to 650 pupils in 18 to 36 classes.

The teaching week is divided into typically 25 to 40 periods (5 to 8 periods per day) and each period is devoted to one of 8 to 15 distinct subjects, with no free periods. Each year group follows a defined curriculum with typically 4 or 5 periods devoted to major subjects such as English or Science, down to just 1 period a week for minor subjects such as Music.

Introduction

A school's staffing is such that in most departments there are just enough teachers to teach each of the forms in one year band at the same time. Thus the timetable is basically planned on a 'band' basis rather than a 'class' basis and for most subjects, all the pupils in a particular band will be taking the same subject in any period. The only exception to this general rule is in the case of departments such as Music or Drama where there may be only 1 teacher available. In these cases, that subject is often subsumed within one of the major subjects such as English and one of the classes in any band is taught Music (say) rather than English on a rotational basis.

Timetabling is often the responsibility of a senior teacher (a deputy head teacher or a head of department) who does this manually each term, usually on the structure of the previous term's timetable. However, occasional changes to the curriculum and more frequent changes of staffing (particularly the use of part-time teachers) mean that the timetable can seldom be carried over from the previous term or year ; and very often the changes are such that the whole timetable has to be virtully redesigned as only a limited amount of 'tinkering' can be done before the timetable begins to fall apart. With this in mind a timetabler often finds it very difficult to make alterations to satisfy changes that may arise during the school term and when a timetable has been set, changes cannot usually be entertained until the following term or year. A computer approach to solving these problems would be very much welcomed by most timetablers.

The problem presented by the sort of high school described above is essentially to timetable each of the year bands (within which there may be 3 to 4 different classes with the same curriculum requirements) over typically a 30 period week in accordance with a curriculum requirement for up to 15 subjects with prescribed frequencies (numbers of periods) for each subject. These subjects vary according to the way the timetable is set up. In some schools, subjects like Humanities and English are combined to form one subject in timetabling. Staffing considerations

## Introduction

dictate that any subject can only be taught to at most one year band during any period. It would also be desirable if the number of occasions when any subject is taught more than once on the same day (apart from prescribed double periods) to any year band is minimised, and teacher free periods are spread as evenly as possible over the week. These last two soft requirements are likely to be positively correlated in this instance, as spreading the lessons in any subject throughout the week will very probably lead to the same feature for the teacher free periods.

## 1.4 What makes timetabling difficult ?

Timetabling problems are difficult to solve due to the inherent combinatorial complexity and the constraints which form an important part of the problem itself.

### 1.4.1 NP-Completeness

A problem is said to be a Non Deterministic Polynomial (NP)-complete if no polynomial-time-bound algorithm has been found for that class of problem. It is known that the timetabling problem is an NP-complete problem (Evan *et at.*, 1976). This has been shown to be the case even for a simple and naive teacherclass model which ignores the realities of a normal timetabling problem. Since the restricted timetabling problem is shown to be NP-complete, the wider timetabling problem is also NP-complete.

Timetabling is often tackled as a problem of search where, given an initial situation (usually not feasible), search techniques can be used to reach some goal (a feasible or improved timetable). The search space is large and cannot be explored exhaustively for problems of any realistic size; indeed the space increases exponentially as the size of the problem increases.

Introduction

## 1.4.2 Constraints

A number of requirements, often known as constraints, must be considered when drawing up the timetable. The more obvious/essential constraints are

- The timetable must avoid all conflict, so that no student is faced with having more than one lesson at the same time and no teacher having to teach two classes in the same period.
- Every subject stated in the curriculum requirement matrix must be assigned in full.
- Certain periods, such as tutorial periods or assemblies, are common to all year groups and must occur at the same time on all timetables.

In real life timetabling, however, many more constraints may be involved, such as

- Not all classes last the same amount of time. This might range from double period being devoted to, say a craft subject, to a full afternoon for games. In most situations, all classes are multiples of the basic period, but in some cases, classes of a totally different time might have to be incorporated.
- No account is taken in the basic model of teaching or ancillary space. Realistically, it can usually be assumed that there are enough teaching rooms available in total to accommodate all groups of pupils, but there will inevitably be problems caused by the use of special rooms such as laboratories or workshops. These problems become acute where facilities are shared by a number of teachers as would be the case with a gymnasium which is used by more than one PE teacher.
- A number of classes very often have to take place either at restricted or prearranged times.

#### Introduction

### Chapter 1

• A certain period or range of periods may be forbidden for a particular subject. This could be because that subject is taught by a part-time teacher who is not available at that time. Alternatively, certain subjects (e.g active subjects) should not be assigned consecutively.

The constraints listed above are by no means an exhaustive list of such constraints, but they will at least serve to illustrate some of the difficulties that can be expected to arise in practice. Even the simplest educational timetable is likely to involve a number of additional constraints beyond those mentioned above.

## 1.5 Organisation of The Thesis

The organisation of this thesis is as follows.

Chapter 2 reviews the relevant literature of the general timetabling problem. It is divided into a review of timetabling problems in schools, timetabling problems in other organisations, a review of some systems which have been built to solve timetabling problems and finally a review of the most recent methods of approach for solving timetabling problems.

Chapter 3 concentrates on methodology and definitions. It begins with several definitions of the problem and the systematic approaches to the building of a timetabling system. A brief outline of the method used in the following chapters is also given and this includes an evaluation of the potential of heuristic methods for solving timetabling problems.

Chapter 4 considers a heuristic approach using tabu search for solving timetabling problems. This chapter commences with a description of the procedure for

#### Introduction

generating an initial timetable, from which tabu search is used to generated a feasible timetable. Tabu search is then used to satisfy double period requirements and also to produce well-spread subject distributions over a week.

Chapter 5 provides an alternative approach to arriving at a feasible timetable using a simulated annealing approach. This chapter commences with a review of simulated annealing and its use in solving timetabling problems. A general framework of simulated annealing is presented. A framework for solving timetabling problems based on swap operations between periods and cells in the matrix is presented. A comparison of the use of heuristics based on tabu search and simulated annealing is included.

Chapter 6 describes the use of rule based systems to assist in improving the quality of the timetable. This chapter commences with a review of artificial intelligence and expert systems followed by a discussion of the necessary elements in building an expert system. Several soft constraints are defined and heuristic models for satisfying individual and paired constraints are built. Rules are designed to implement each of the necessary constraints which may be specified by the users.

Chapter 7 describes the implementation of the three approaches into a timetabling system known as ACT (Automatic Computerised Timetabling). This system was tested with timetablers from three chosen schools which serve as case studies. The system is carefully designed so that it can be used instantly by timetablers. Several of the routines are described in this chapter.

Chapter 8, the final chapter of this thesis, contains a discussion of the overall work relating to timetabling problems. This chapter gives a summary of the thesis, a discussion of the results obtained and suggestions for future work in this area.

Literature Review

# CHAPTER 2

## **REVIEW OF THE LITERATURE**

## 2.1 Introduction

The process of constructing a timetable takes place again and again on a regular basis. This has led many researchers in this field to attempt to create computer based tools for assisting timetablers to generate an acceptable timetable. This research, as mentioned in the previous chapter, focuses on heuristic models and the use of a rule-based approach to develop an interactive computer program to solve typical timetabling problems in high schools. This chapter commences with a discussion on the role of computers in timetabling and early computerised systems in schools, followed by a discussion of several methods used in solving the problem.

## 2.2 The Role of the Computer in Timetabling

Computers can play an immensely valuable role in timetabling, especially in areas such as recording, storing and checking data; manipulating this data in the construction of a timetable; cross checking the stages of construction; and also in the considerable task of printing out the many results. It is generally agreed that the main role of computers is to aid the second stage (see Section 1.2) of timetabling, that is the detailed construction of a timetable from a school's basic requirements. The benifits of using computers in timetabling can occur in the following areas,

• It will save the timetabler weeks of tedious effort, freeing up time for more important tasks.

- It may encourage more frequent timetabling. Without a computer, many schools leave their timetables unaltered for long periods of time. The computer, will allow schools to timetable more than once a year if necessary.
- It can provide a detailed check of the feasibility of the timetable requirements, allowing the timetabler to experiment with different forms of curriculum organisation to see which he likes best.

For the past two decades, there have been many approaches to solving timetabling problems using computers, some much more successful than others. Computers are used in solving university timetabling, college timetabling, examination timetabling, seminar timetabling and the timetabling of meetings among managers in large organisations. This thesis will concentrate on computerised timetabling in schools with multiple constraints, designing heuristic models to generate a timetable for high schools, and demonstrating the use of rule based systems in the construction of a timetable. In this chapter, a description of various systems that have been developed and used successfully in schools is given. Section 2.3 describes computerised timetabling in schools and provides a brief description of the systems tried or used in schools. The chapter continues with a description of the techniques used to produce a timetabling program that have been tried in various establishments.

## 2.3 Early Computerised Timetabling in Schools

This section provides a brief description of the early development of computerbased school timetabling systems. Several authors have described the complexity of the problem and introduced different approaches to tackle the problem. An early method for using a computer to produce a timetable for a medium sized school was

#### Literature Review

introduced by Appleby et al. in 1961; a system which took more than an hour to produce one feasible timetable. His work was then extended by Gotlieb (1962) whose work became the basis of many other methods. Gotlieb identified tight class allocations and introduced dummy classes and dummy teachers into the standard form to ensure that the number of classes and teachers are equal for each period of the week. This approach was successfully tested but the time to generate a feasible timetable was just too long. This work was then extended by Csima (1965) and Lions (1966; a, 1967; b, 1971). They treated the problem more rigorously as one of combinatorial mathematics and replaced the heuristic approaches proposed by Appleby and Gottlieb with a mathematical structure. This system split the weekly problem into the appropriate number of daily problems so as to achieve a desirable spread of subjects over the week. The actual rules and a number of theoretical contributions to the operational system were developed by Lions (1966a) and Griffith (1966). The implementation of this system was not very successful as it produced inconsistency in the result. This inconsistency would not be detected until much later in the algorithm, by which time there is nothing to do but go back to the beginning, shuffle the input data, and re-run the program, hoping it would not make the same mistake again. This procedure was successfully tested but its running time to generate a solution was again too long.

Baraclough (1965) produced a method of constructing a school timetable using a computer which suggested that the manual approach be adopted where appropriate and supplemented where necessary. He also gave some criteria for judging the success of the timetable produced, along with a theoretical discussion of the different methods of allowing a choice of sixth form subject. The Local Government Operational Research Unit (LGORU), with the aid of the National Council for Educational Technology, carried out an extensive test in the UK. Baraclough's method of timetable construction includes both combinatorial methods and heuristic rules containing some intuitive elements such as higher

priority requirements which are made to take precedence over those of lower priority. Some of the results are demonstrated in various systems such as the Teddington System.

## 2.3.1 The Teddington System

A method proposed by Appleby, Blake and Newman (1961) led to the development of the Teddington System. This system is based on two heuristic rules that expect the timetabler to (i) start by scheduling those requirements which have the least scope for manoeuvre; and, where there is some degree of choice, (ii) assign a requirement to a period in which it is likely to interfere minimally with other requirements of the timetable. The input to the program is a number of lines stating that a given class must meet for a stated number of single and double periods. A square conflict matrix, whose dimension is the number of lines, is constructed, with a non-zero entry indicating that the corresponding lines are involved in one of the above special requirements. Each line is then given an availability vector stating which single and double periods of the week are available to that requirement. Starting with a blank timetable, requirements are scheduled according to the following rules:

- Select the requirement where the difference between available and required periods is minimal.
- Where the difference between available and required periods is positive (i.e. there is some choice) a period is chosen to minimise interference, a measure which takes account of the reduction in availability to all conflicting lines caused by choosing that period.
- When a period is assigned, the availability of all conflicting requirements is reduced appropriately. If at any stage the number of periods available is less than the number of periods required, the program has failed.

#### Literature Review

This system is capable of producing a timetable for a *medium-sized* high school; a process which takes about one and a half hours to produce one feasible timetable. Although the program was not really operational, the ideas were important to the development of later systems.

## 2.3.2 The Oxford System Group

The work of the Oxford System Group (OSG) is an extension of Gotlieb's procedure and is also based on the work of Dempster (1968) which developed the concept of interchange. The OSG approach is described as follows. "Instead of starting with a blank timetable and inserting each requirement in turn, checking for an unfeasibility at each stage, it takes a timetable which satisfies all the simple requirements, and modifies it to include the pre-assignments." (OSG, 1970). The initial timetable could be last year's timetable or one that is generated automatically from the requirements matrix. This process of altering a complete timetable is known as interchange and Dempster (1968) derived necessary and sufficient conditions for such an interchange to be feasible.

### 2.3.3 The Nor-Data System

In 1966 a school timetabling system devised by Michalson began field trials in Norway and by 1970 his Nor-data system was producing 100 schedules each year. The Nor-data system begins by checking all the necessary requirements and after all timetable checks have been completed, the information is transferred to a data table. The process in this system is divided into three parts

• The data is checked for errors (for example period totals for classes and teachers),

### Literature Review

### Chapter 2

- Check for timetable infeasibilities. If found then regenerate the initial table and
- Continue scheduling.

This system always produces a schedule but there is a problem in some cases that the schedule needs to be modified manually if the derived solution is not suitable. The manual adjustment very seldom took more than 8 hours for any of the examined school structures.

## 2.3.4 The Strathclyde Program

In various publications, Lawrie (1968,1969,1975) described the approach used in the Strathclyde Program. This program was developed for a particular group of schools, and consists of both theoretical and practical aspects. The data are presented in the form of tables which are called layouts. The input was limited to a relatively small number of layouts and since the layouts could not describe all of the school's requirements, it did not produce a complete timetable. The implementation of this system revealed that the computing could take a long time. The system was not widely implemented, but it did demonstrate the possibility of using integer linear programming, and that it was possible to demonstrate that a particular problem is infeasible.

Lawrie and Veitch (1975) in their book *Timetabling and Organisation in Secondary Schools* gave a coherent account of some manual and computer-based methods for timetabling in secondary schools. The layout method, its advantages and disadvantages, were discussed and they also gave the data requirements of the Strathclyde procedure and the problems that arose from various field trials.

## 2.3.5 The SPL System

This system was developed by Dr. Charles Kent of Systems and Programs Ltd. This system was later introduced in the United Kingdom but it did not receive much attention. The SPL system uses a mix of combinatorial and heuristic rules selecting periods for requirements to satisfy constraints which include subject distribution through the week and satisfying staff requirements. The basis of the SPL system is that it begins with a blank timetable and proceeds to fill it in by calculating a degree of freedom for each item, selecting the item with the least degrees of freedom and searching for positions to allocate it. For each position that is found, the cost of allocating it there is measured and the position with the least damaging effect is chosen.

#### 2.3.6 The SIMS Approach

This is a School Information Management System produced by The Local Education Authority (LEA) SIMS Support Team based in Dunstable, England. The main elements of the system are the curriculum system and the resources management system (Finance/Accounting). The curriculum system provides course allocations (that is it allocates students to optional courses and writes the information automatically to the student record), staff cover scheduling and school report writing. The resources management system provides budget planning, financial monitoring and control, and a library management module. This package is more of a management package than a package used to solve timetabling problems.

### Literature Review

#### Chapter 2

## 2.3.7 Other Systems

Other systems, such as the Price Waterhouse System (PWS), also used combinatorial mathematics and heuristic approaches. This system had very limited success. Similarly the Scicon System, developed by Millane (1972), which uses a network analysis computer program, had a poor response from schools.

Dempster and de Werra (1975) suggested the use of graph theory to solve the timetabling problem. Their work is based on network flow and they later developed a computer procedure which proved quite successful when applied in several Swiss schools. Their procedure deals only with the allocations of single period meetings between one class and one teacher, so that setting, double periods and the allocation of classes involving specific rooms must be *pre-assigned* prior to using the computer. Mulvey (1982) showed that graph and network models are efficient to use even for a large-sized problem. When applied to real timetabling problems, however, this approach failed to solve the problem using purely network models.

Since the mid 1980's much more work on the subject of computer timetabling has been carried out but mostly in the form of mathematical modelling and network flow models. de Werra (1985) used both graphs and a network approach in the formulation of a timetabling problem and suggested several other models, with the emphasis on graph theoretical models. He described an interactive system based on these models which is used to produce feasible timetables in a much more efficient way than his earlier method.

In the following section, several newer techniques for solving timetabling problems are discussed.

## 2.4 Recent Techniques for Solving Timetabling Problems

In recent years, new search techniques have been developed and applied to solving timetabling problems. These techniques have involved simulated annealing, tabu search, neural networks, genetic algorithms, artificial intelligence and expert systems, logic programming, graphical methods, and spreadsheet and database approaches. Some of these methods and their implementations are summarised in Table 2.1 and each of these techniques will be discussed in the following subsections.

Technique	Authors	Implementation
Decision Support Systems	White and Wong (1988);	Schools
	Chahal and de Werra (1989);	Schools
	Tripathy (1992)	University
Spreadsheet Approach	Murphy (1987)	University
Graph Colouring	de Werra (1985)	University
	Selim (1988)	University
Expert System	Dti Team (1990)	Rolls-Royce and
-		Associates:
Logic Programming	Monfroglio, (1988)	Schools
· · · · · · · · · · · · · · · · · · ·	Kang and White (1992)	University
Meta-Heuristic Approach		
• Simulated Annealing	Abramson (1992)	Schools
	Johnson (1990)	University
Tabu Search	Hertz (1991;1992)	University
Genetic Algorithms	Abramson (1992)	Schools
Neural Networks	Kovacic (1993)	Schools
	Abramson&Abela (1992)	Schools
Database Approach	Johnson (1993)	University

TABLE 2.1 The techniques proposed by different authors and places where the respective techniques were used for solving timetabling problems.

#### Literature Review

### **2.4.1** Decision Support Systems

A number of timetabling problems are heavily constrained and some of the constraints are in conflict with each other making a feasible solution impossible if all the constraints are to be satisfied. Objective functions have been used extensively to overcome this problem by balancing the satisfaction of the different constraints.

The use of interactive programs for timetabling (White and Wong (1988), Chahal and De Werra (1989) and Tripathy (1992)) has given users a tool with which to control the process and give it a human touch. There are several aspects to this interaction.

- Constraint relaxation, whereby certain constraints are not used any more or are given lower weight.
- Modification of initial data in order to relax some initial requirement.
- Giving the user the opportunity to try out different problems by altering initial information.

### 2.4.2 Spreadsheet Approach

Murphy (1987) attempted to solve a school timetabling problem by combining a rule-based approach and the interactive capabilities of a spreadsheet. Class-teacher-room combinations are assigned to periods and the system requires the user to make the assignment. The system keeps track of the constraints and helps to maintain a clash-free timetable.

A two dimensional representation in the spreadsheet represents the teacher (i.e. column) and the period (i.e. row). Each cell holds information about the class,
#### Literature Review

room and subject. Rules are used to ensure that the assignments are made without any clashes among the resources involved. As well as assigning a lesson, the user can reassign a period (i.e. undo), complete a particular period with assignments, find an empty room in a particular period, and jump to another area of the spreadsheet.

This system may be useful for a small school where the user has the opportunity to play around with the timetable and look at different possibilities. In a large school this might not be very practical.

### 2.4.3 Graph Colouring

de Werra (1985) gave a description of a graph colouring algorithm. It is based on assigning teacher-class combinations to periods such that there are no clashes of resources in the timetable generated. The algorithm enables the changing of colours for the vertices, which could be a change of period for a particular lesson.

The timetabling process starts with a complete timetable where all the lessons are given a timeslot but not all constraints are satisfied. The timetable could be from the previous year, which needs some modification due to changes in the requirements. The process works by re-labelling a vertex which induces changes in other vertices due to an incompatibility. This could be done recursively until a point is reached where there is no incompatibility in the graph. The shifting which occurs during the re-labelling is(to) referred by the author as "interchange". Interchanges could be used to avoid clashes and accommodate other lessons into a timetable. This program is tested successfully using some hypothetical data. Preassignment is considered, but other special requirements (e.g. double periods)

#### Literature Review

#### Chapter 2

were not considered. This is basically a theoretical work and the practical application of it is not reported.

Selim (1988) described a technique which could reduce the chromatic number of a graph. This is for a faculty timetabling problem which attempts to give the students as much choice as possible in choosing the courses they take. It is shown that a vertex could be split by replacing it with two or more vertices which would make up the original vertex. Some of the original edges which were incident to the original vertex will now be incident to the newly split vertices. The author uses examples from his university, where a big class of students can be split into smaller groups according to their major (i.e. degree for which they are enrolled). These smaller groups have less conflicts (i.e. less edges) with other courses compared with the original big class of students. This leads to a small chromatic number which means less periods are required. This is again a theoretical work, but a solution was obtained for a faculty at the author's university.

# 2.4.4 Expert Systems

Research at Rolls-Royces and Associates Limited (Dti, 1990) has produced an expert system known as MEEPLES to schedule meetings among the managers in that company. The aim of this system is to maximise attendance at meetings and to ensure that the correct flow of information is maintained by appropriate spacing between meetings which resulted in an improvement in the quality of meetings. Before this system was used, the key participants could not attend all the scheduled meetings and had to be substituted by someone less well-informed or not having the same level of authority. In implementing this system, several requirements must be provided which can be characterised by the following factors:-

• fixed frequencies of project reviews and hierarchical flows of information

Literature Review

- availability of resources (personnel and rooms)
- imposed constraints or preferences as declared by the chairman.

This system works from a set of input data which is prepared by managers on special forms which are then processed by the planning division. These sets of data will go through two phases; the first phase is the plan generation and the second phase is the plan repair. The system performs most of its timetabling automatically, but on occasion it needs some help from the planning department users.

The knowledge in this system is about constraints and preferences for meetings, and how to relax the constraints if an ideal situation cannot be achieved. An example of a simple rule used in finding a suitable allocation may be as follows:

IF the number of replacements is greater than the maximum allowed,

THEN alter the start time of the meeting.

These meetings form a natural hierarchy and the search exploits this in a 'topdown' fashion.

The user interface allows for the input of a large amount of meetings requirement data, user intervention to steer the scheduling process, and explanations. Meeples was implemented on a Sun 3/60 workstation with 16 Mbytes of memory. Even though it managed to produce a working timetable, its average execution time is about three hours.

# 2.4.5 Logic Programming

A logic programming technique using PROLOG is reported in Monfroglio (1988). The system assigns teacher-class-student combinations to periods. A knowledgebase of assertions, inference rules and relations is constructed. This includes the

#### Literature Review

#### Chapter 2

domain facts and rules, the heuristic and strategy meta-rules, the planning metarules and the truth maintenance rules. The knowledge is represented using a production system which is non-monotonic and non-communicative. The system uses three levels for the problem solver:

- the domain level (e.g. databases for students, teachers and rooms)
- the method level (i.e. choice between two strategies for timetabling)
  - the priority for each class
  - the priority for each room
- the planning level (i.e. finding a suitable plan)

Kang and White (1992) studied constraint resolution with hard and soft constraints examined in the context of generating a university timetable. They based their work on a logic programming approach and it was implemented using Waterloo Prolog (W-PROLOG). Logic programming was used to represent the constraints while pattern matching and backtracking facilities were used for the search process. They took this approach because of modularity (i.e. the program can be divided into independent modules) and feasibility (a large number of complicated constraints are more easily formulated within the first order logic than they are using other, more common techniques because they do not require any procedural specification).

The system differentiates between generic and specific resources, and attempts to optimise the utilisation of specific resources. Generic resources are considered to be *less difficult*. The truth maintenance system used by the system uses a simple mechanism called reinforced logic where a fact is true if it is reported as such and no negation is reported. The fact is false if a negation is reported.

### Literature Review

### Chapter 2

The timetabling problem is decomposed into two sub-problems; time scheduling where a class-teacher (i.e. lesson) pair is assigned to a period, and room scheduling which assigns rooms to lessons. The hard constraints have to be satisfied while the soft constraints could be relaxed to obtain a solution. The soft constraints are given priorities to assist in the relaxation process.

In room scheduling, the rooms are categorised according to their sizes and given a range number. A room which meets the exact requirements of lessons may not be available but a slightly bigger room may be free. Although this is a multi-level program, it does not provide facilities for going back and forth between the stages. Such a facility would give an opportunity to modify a solution if it is found to be unsatisfactory at a latter stage and could not be modified locally.

A course scheduling system which is implemented using PROLOG is reported by Gunasena *et al.* (1992). They attempt to solve a course to period assignment problem. The proposed algorithm selects the next course to be scheduled from a list of courses and the next timeslot which is suitable for this course from a list of timeslots. If no timeslot is suitable then the algorithm backtracks to the previous course to undo it and find a replacement timeslot. This backtracking procedure is based on chronological order and in this implementation chronological backtracking is not very successful in those cases where the search space is very big. The system was implemented on departmental data but information provided is too sparse to draw any real conclusions.

# 2.4.6 Knowledge-Based Approaches

Grudes et. al., (1990) gave a general paradigm for solving resource allocation problem using an expert system approach. It is implemented for a flight crew

#### Literature Review

assignment problem and a timetabling problem using PROLOG. The paradigm has three major components:

- 1. A common set of concepts to define most resource allocation problems.
  - Activities tasks which must be performed.
  - Resources needed to perform the activities.
  - Priorities of the activities and resources to direct the allocation.
  - Allocation assignment of resources to activities.
  - Constraints restrictions on allocations.
- 2. Rules for the allocation procedure.
  - Restrict rules (i.e. hard constraints) which must be satisfied.
  - Recommending rules (i.e. soft constraints) which are satisfied where possible.
- 3. Strategy for resource allocation.
  - forward allocation; finding the most difficult unassigned activity for allocation purposes.
  - consistency checking; validating the allocation with the relevant rules.
  - backtracking by local changes; in case of failure to allocate, it is necessary to backtrack to a previous assignment so that it can be deallocated to allow the current activity to be allocated.

The timetabling implementation involved the courses as the activities and the periods as the resources. The course definition included all the necessary components i.e. teachers and students. Priorities were decided on the basis of difficulty in allocating a course. The recommending rules were a general guideline on how to allocate a certain course or a group of courses. The constraining rules were the restriction on the component of the activities relating to the resources. Candidates for local changes should be courses with other choices of periods.

# Literature Review

#### Chapter 2

The forward allocation is done on the basis of the most difficult course being allocated first. This allocation is checked for consistency against rules for the allocation procedure. If an allocation cannot be made, local changes using backtracking are carried out. This is basically dependency directed backtracking whose success depends on the tightness of the situation.

# 2.4.7 Heuristic Approaches

# 2.4.7.1 Heuristics based on Tabu Search

Tabu search originated in solving large non-linear combinatorial optimisation problems, and has been applied subsequently to a diverse collection of problems. It was introduced by Glover (1977, 1986) and was first implemented under the name of the oscillating assignment heuristic. Glover (1990) provides an introduction to the principle of tabu search and gives some examples of its successful application. A brief explanation of the ideas behind tabu search is presented in this section.

Tabu search is known as a *meta-heuristic* procedure; that is a heuristic method where, at each step, another heuristic local search procedure could be used.. This is a process which begins with a feasible solution and, whenever there is a feasible solution, a list of other solutions called a *neighbourhood* is generated. The aim is generally to move from the current solution to one of the neighbourhood solutions which leads to an improvement in the current solution.

The feature that characterises tabu search is the presence of *forbidden moves*, having attributes which make them tabu. Tabu search is sometimes known as a constrained search technique because each step of the process involves moves that are not excluded by the current tabu conditions.

Tabu moves have the goal of preventing cycling, and those moves that are considered tabu are placed in a list known as a *tabu list* or a list of *forbidden moves*. This tabu list can be determined in a variety of ways, but the most frequently used criterion is one of recency where the tabu list includes the most recent solutions. Deciding that some moves are tabu will generally reduce the choice of moves. An additional feature is needed to cancel the tabu status of a move when this move gives considerable improvement. This is often known as an *aspiration function*. This aspiration function should be compatible with the goal of avoiding cycling, while increasing the likelihood of an improved solution. A move is called a *permitted move* when it is either not on the tabu list, or is on the tabu list but satisfies the aspiration criterion. The role of the aspiration criterion in tabu search is to allow tabu status to be overcome if a move is sufficiently attractive. A typical aspiration criterion would be if the move results in a solution which is better than the previous best solution.

Tabu search can be used to solve a variety of search problems from employee scheduling (Glover and McMillan, 1986) and machine scheduling (Laguna *et al*, 1991), to graph colouring (Hertz & de Werra, 1987), graph clustering (Jaumard *et al.*, 1992) and travelling salesman problems (Malik *et al*, 1989). The search has the ability to employ any procedure as a subroutine in order to arrive at a local optimum. It has the ability to escape from a local optimum, thereby increasing the chances of reaching the global optimum.

Hertz (1991) considers the use of tabu search for solving a large school timetabling problem. In the case reported, he uses an initial timetable which does not have any conflicts except those involving teachers or students having more than one course at the same time, or having to move from a distant building. In searching for a

### Literature Review

feasible solution, reassignment of courses is directed towards those courses which reduce conflicts. This avoids neighbours which do not improve the objective function. The algorithm was implemented to solve a timetabling problem which involved 288 single section courses (i.e. lessons per week), 143 teachers, 67 rooms and 1729 students. The second application in this report was on an examination scheduling problem where about 20% of examinations were preassigned. In both applications, the schedules produced were regarded as successful.

Hertz (1992) uses tabu search to solve course scheduling problems in which the lengths of courses are not necessarily known in advance. The approach began by considering a model for class-teacher timetabling followed by course scheduling. In formulating the problem, topics are divided into static topics (which are fixed in advance) and dynamic topics (which can be altered). Several constraints were listed and each constraint is dealt with one in one of the following ways: it can either be satisfied by one of the set of feasible solutions or else it can be penalised in the objective function. A decision has to be made as to which requirement will always be satisfied and which will be penalised. In this implementation, hypothetical data from a school were used to test the efficiency of tabu search.

## 2.4.7.2 Heuristics based on Simulated Annealing

Simulated annealing is a technique which can be used to find (approximate) solutions to combinatorial optimisation problems. The central idea of this technique in solving a minimising problem is that certain *uphill* steps may be required to prevent a local search scheme from getting stuck in a poor local optimum. In the most common version of simulated annealing, *downhill* (that is an improvement) perturbations of the existing solution are always accepted but an uphill step is only accepted with a specified, and decreasing, probability.

# Literature Review

Historically simulated annealing originates from a combination of ideas from two fields, that is the analogy between the physical annealing process and the problem of finding (near) minimal solutions for discrete minimisation problems. The physical annealing process is a *thermal* process for obtaining low energy states of a solid in a *heat bath*. The technique simulates the cooling of a collection of hot vibrating atoms. When the atoms are at a high temperature they are free to move around and tend to move with random displacements. However, as the mass cools, the interparticle bonds force the atoms together. Metropolis *et. al.*(1953) proposed a method for computing the equilibrium distribution of a set of particles in a heat bath using a computer simulation method.

The given state of a solid having energy  $E_1$  is compared to the state that is obtained by moving one of the particles to another location by a small displacement. This new state, with energy  $E_2$ , is accepted provided that  $\Delta E = E_2 - E_1 < 0$ , i.e., at any given temperature, a new configuration of atoms is accepted if the system energy is lowered. If  $\Delta E > 0$ , the new state is generally rejected, but occasionally the change is accepted with a predetermined probability  $\exp(-\Delta E)/(kT)$ , where k is the Boltzmann constant and T the temperature of the *heat bath*. This probability is governed by the parameter T, which decreases the probability as the process progresses. So a move to a state of higher energy, a *worse state*, is accepted in a limited way. The rate at which T is reduced is critical to the success of the method and is known as the *cooling schedule*.

The method described above was used by Kirkpatrik *et al.* (1983) for solving hard combinatorial optimisation problems. They replaced the energy by a cost function, and the atomic state of a physical system by a solution of a combinatorial minimisation problem. The perturbation of the particles in the physical system then

#### Literature Review

becomes equivalent to a trial in the combinatorial minimisation problem. The minimisation is done by first *melting* the solution space at a high temperature (temperature now being simply a control parameter), and then slowly lowering the temperature until the system is *frozen* into a stable solution. Many optimisation problems can be considered as a set of activities which need to be scheduled such that an objective function is minimised. The vibrating atoms are replaced by the activities and the value of the objective function replaces the system energy.

Essentially, simulated annealing involves defining a neighbourhood structure on the set of feasible solutions and then moving from the current solution to a randomly selected neighbour. If the new solution is an improvement, the move is made. If the new solution is worse, then the move is accepted according to a predetermined probability distribution. This probability is governed by the temperature parameter, which decreases the probability as the process progresses. Eventually the process stops after satisfying a set of stopping criteria. Aarts, Korst and Laarhoven (1988) gave a review of the theory and application of simulated annealing.

Abramson (1992) attempted to solve a school timetabling problem based on simulated annealing and using real data from an Australian high school. The application of simulated annealing to timetabling problems replaces the atoms by a particular combination of a teacher, a subject, a room and a class and system energy by timetable cost. Timetable cost is generally the sum of a class cost, a teacher cost and a room cost, each of which is the sum of the number of clashes involved. At each iteration, a period is chosen at random and a lesson is chosen at random from that period. Another period is also chosen at random. The cost of removing the lesson from the first period and the cost of inserting it at the second period are calculated. Since simulated annealing relies on random choice of periods and lessons it cannot guarantee that the true minimum cost value is actually found or that two different annealing runs will yield the same value. The run is terminated if

the cost becomes zero or the cost has not changed for a certain number of iterations.

Several practical problems associated with using simulated annealing for simple timetable construction were identified. It cannot handle classes which have students in common, it is not flexible in assigning rooms to elements, it does not allow multiple periods, it does not allow one class to be scheduled always with another class and it does not allow one type of clash to be more important than another.

The results of the serial algorithm are quite promising but the time taken to generate a timetable using the high school data was quite long. A parallel program was introduced to improve the speed and was reported to have some successful results but with some problems of implemention.

# 2.4.7.3 Heuristics based on Genetic Algorithms

A genetic algorithm is a procedure that imitates the process of natural selection which occurs by mutation of genes in chromosomes. The process tends to promote the more desirable genes and does not pass on the less desirable ones. It is a process where the fittest genes survive increasing the average fitness of a population in each generation. A general description of genetic algorithms is contained in Goldberg (1990).

Abramson and Abela (1992) used this method in their attempt to solve a school timetabling problem. The main issue in timetable construction is to avoid clashes and a cost function was used to remove these clashes. It consists of the number of clashes in the timetable and also includes a teacher cost, a class cost and a room

cost. The optimization problem is to minimise the total cost and an acceptable timetable is one with zero cost.

Colorni *et al.* (1991) also attempt to solve a timetabling problem using genetic algorithms but do not provide much detail of the algorithm. The objective function used considers clashes and the spread of lessons over the week and also some organisational and personal goals.

# 2.4.7.4 Heuristics based on Neural Networks

A neural network is a procedure that imitates the behaviour pattern of the brain which consists of large numbers of neurons, inter-connected by synapses. One feature of the neural network architecture is that the processing units are analogous to neurons. These processing units are often known as nodes. Nodes are connected by directed links with associated weights. These weights are obtained by a training process which can be either supervised or unsupervised. Neural networks have been used for solving many combinatorial problems and Kovacic (1993) attempted to solve a timetabling problem using this technique. He defined two operators for searching the state space; the first is to move a subject from its current position in the timetable to another position and the second is to swap two subjects.

In modelling the problem, a timetable is presented as an  $r \ge p$  matrix where r is the number of rooms and p is the number of periods in the timetable, with the subjects as the matrix elements. Various constraints are given and a cost function is optimised to 0. An optimal solution is obtained if all the subjects are filled in the timetable in such a way that all the constraints are satisfied. The application of Markovian neural networks for a timetable has s neurons (subjects), one neuron for

#### Literature Review

every subject to be scheduled. Every neuron has  $r \ge p$  states which correspond to possible entries in the timetable and the state of the network represents a timetable (which may or may not be accepted). The initial swapping of neurons is selected at random. In every iteration only one neuron is allowed to change its state and the selected neuron determines the objective function for the change from one state to another state according to the probability of transition. The algorithm was implemented to solve a timetabling problem involving 37 teachers, 40 periods, 33 rooms and 202 subjects. The schedules produced appear to have been successful.

# 2.4.8 Database Approaches

Johnson (1993) proposed an approach using databases for solving timetabling problems in an educational establishment. The proposed system is designed to be used mainly for the *book-keeping* tasks in timetable construction. The timetable information is stored in one or more files containing information such as course code (which identifies a distinct series of classes taken by a lecturer and attended by one or more groups of students), subject/area, teachers, day of the week, time of the day and room. Besides these primary data files there are other secondary files which are used for cross-referencing in the production of timetable information.

The implementation of this system has proved to be successful and it is able to reproduce term timetables for more than 500 students, on 5 separate degree programs and 80 members of teaching staff. It also has the facility of printing several types of reports and forms which are tailored to the requirements of Loughborough University Business School. It was suggested that improvements to the system could be achieved by using an expert system based on knowledge from the previously generated timetable. This could be another new research area to explore the possible links between data base and expert systems.

# 2.5 Summary

Upon reviewing the problem, it is found that timetabling has been formulated mainly as a feasibility problem (finding a feasible solution) rather than an optimality problem (finding an optimal solution). Generally timetabling problems do not have a well-defined objective function to optimise; there are many requirements which occur as constraints in the problem and a collection of wishes which are not always completely explicit. Even though some of the literature reviewed here is quite old, the problems described are similar to the current situation and this review is included for a better understanding of the problem. Main interest from this review focuses on three main approaches, that is on tabu search, simulated annealing and the possible use of rule-based expert systems in constructing a feasible timetable.

Mathematical formulations of the problem using methods such as integer linear programming have not been generally successful and many authors have moved towards developing heuristic methods. A heuristic method usually entails assigning the subjects with most difficult constraints first followed by those subject with less difficult constraints. In resolving conflicts, it appears that shifting and swapping (transferring position) are the two most common operations used, but this needs to be done efficiently. These operations will be used extensively in developing heuristic models in this thesis particularly in rule based approaches and interactive operation. As suggested by Johnson (1992) there is probably some benefit in using expert systems and rule based approaches as implemented in Meeples, particularly in conjunction with appropriate heuristic techniques. This thesis will explore such an approach.

Background Work

# CHAPTER 3

# **BACKGROUND WORK**

# 3.1 Introduction

This chapter presents the direction of the thesis and an overview of the methods used. It begins with the research aim and a brief account of the nature of the timetabling problem encountered in high schools, which form a basis for the development of the proposed computerised timetabling system. A discussion of the timetabling process is given and a brief comparison is made between the heuristic methods based on tabu search and simulated annealing, followed by a description of rule-based expert systems. Finally, a brief background to the design of an interactive Automatic Computerised Timetabling (ACT) System is given.

# 3.2 Research Aim

The aim of this thesis is to explore various approaches to computerised timetabling, particularly the possible role and benefits that expert systems can provide in timetable construction. The research paradigm described in Sections 3.3 and 3.4 involves timetable construction for high schools, and the use of three particular schools as case studies. In this respect the proposed research paradigm was exploratory and experimental in its nature, and therefore it was neither feasible nor appropriate to develop a set of formal research hypotheses. The exploratory and experimental paradigms seek to examine the potential and feasibility of expert systems combined with metaheuristic techniques namely tabu search and simulated

#### Background Work

annealing in solving timetabling problems. The inclusion of an expert system into the process is to provide improvement by utilising its flexibility to tackle the more specialised aspects of timetable construction.

Most solution approaches tend to be specific to one particular application and any attempt to implement them in another situation generally requires substantial modification before being used, if indeed they are applicable at all. It is highly desirable to create a system that will require as little modification as possible or no modification at all. To achieve this it was planned that each case study would involve collaboration with the school to determine the needs of each particular setting in timetable construction, and also to incorporate knowledge from the expert (the timetabler). Using this knowledge, rules would be developed and then incorporated into the design of an interactive system based on the methods mentioned above. Finally, with a working interactive system, the system performance would be measured and its benefits to the school evaluated. Tentative conclusions about the benefits that the developed heuristic models and rule based expert systems provide for the construction of a timetable are drawn from the discussion of the case studies.

# 3.3 Nature of The Problem

The creation of a suitable timetable on either an annual or term basis is a recurring problem in virtually all educational establishments. This problem becomes generally more difficult as the size of the institution and the flexibility of the curriculum increases. At one extreme in a small junior school where each class is taught exclusively by one teacher, the timetable is simply a matter of allocating each period during the week to a particular subject in accordance with defined curriculum requirements. This may well be left entirely to the discretion of the

#### **Background Work**

teacher concerned. Alternatively, in a sixth form college or higher education institution, timetables have to allow for large numbers of students to select subjects or courses on a modular basis to make up increasingly varied 'A' level or degree programmes. Whilst there is no guarantee that any desired combination of subjects or courses will necessarily be feasible, the job of the timetabler is to allow as many choices as possible to be satisfied. Computer technology can play an important role in assisting timetablers to solve their problems, and since 1960 many efforts have been made to develop computer programs to provide an aid to timetabling. The word *aid* is important as systems are designed to assist the timetabler to progress towards a solution rather than to provide an immediately acceptable timetable. The task of constructing a system that will serve as a successful tool to assist the timetabler requires a full understanding of the complexity of the problem.

The freedom to incorporate manual adjustments is often desired by the user (who thus feels that he or she can keep in touch with the system), and it is worthwhile to include an interactive element in the system. It is also considered to be a challenge to produce a quality timetable. What constitutes a quality timetable is a matter of opinion. Good distribution of subjects is one possible criterion, such as periods allocated alternately between active and sedentary subjects, or a subject not being assigned consistently on the last period of the day. From the teacher's perspective, consideration may also be given to when teacher free periods are assigned. These refinements in timetable construction are often so subjective that it is sometimes felt to be an area not worth considering. The degree to which the timetable implements school policy as indicated by the curriculum plan is almost always seen as the main priority.

# **3.4 The Timetabling Process**

Many authors have studied this area and suggested various phases in the timetabling process. In a review article, Johnson (1980) split the overall timetabling process into several phases as follows;

- curriculum planning.
- staff deployment analysis.
- timetable feasibility test.
- assignment task.

# 3.4.1 Curriculum Planning

This is the phase where the aims of the school and available resources are considered. A curriculum plan is produced which shows the subjects to be taught to different groups of students and the number of periods for each of these subjects.

# 3.4.2 Staff Deployment Analysis.

This is the phase where the feasibility of the curriculum plan is analysed; this will include checking the subjects offered and the number of periods involved against the available staff.

# 3.4.3 Timetable Feasibility Test.

This is the phase where situations which are impossible to schedule are identified and eliminated. Certain situations will be mathematically impossible and these should be removed from the problem. Some of these situations may not be very obvious and these may remain to be discovered in the later phases. Although some

#### **Background Work**

#### Chapter 3

infeasibilities can be detected, there is no way of telling if a feasible timetable can be produced for a given set of data.

# 3.4.4 Assignment Task.

The first two phases are common in timetable construction but may not necessarily be followed. Usually the real involvement of a timetabler will begin in the third phase, whereas the first and the second phase may be handled by the head teacher and/or a head of department who may or may not be the timetabler. A traditional school timetabling problem is therefore usually identified with the process of scheduling the various classes or groups over a specified period of time (usually a week) subject to various restrictions on the resources involved. Local conditions will often give rise to additional constraints, either medium or soft, particularly when part-time teachers or outside facilities are used. In essence, therefore, the problem is to allocate each period of the week to a particular subject (and by implication a particular teacher) for each class in accordance with pre-specified curriculum requirements, so as to satisfy all of the hard constraints and as many of the medium and soft constraints as possible. In the following chapters, a detailed description of the work related to data manipulation and the solution approach for producing a working timetable will be given

# 3.5 Solution Method

During recent years, the role of optimisation has steadily increased in such diverse areas as engineering, operational research, computer science and communication. In the last 20 years, major results in combinatorial optimisation, the search for an optimum solution to a function of discrete variables, have been obtained. However, even today, many large combinatorial optimisation problems can only be

#### **Background Work**

solved approximately on present-day computers. This is due to the fact that many of these problems have been proved to be NP-hard (de Werra 1985).

A number of approximation algorithms have been developed which can be divided into two categories, namely algorithms tailored to a specific problem (tailored algorithms) and general algorithms applicable to a wide variety of combinatorial optimisation problems. Two of the more frequently used general algorithms are tabu search and simulated annealing. Tabu search is based on systematic search while simulated annealing is based on a random search process. However, both tabu search and simulated annealing incorporate a number of aspects related to *iterative improvement* algorithms or, as it is sometimes known, *neighbourhood search*.

An iterative improvement algorithm can simply be described as follows. Starting from a given solution, a sequence of iterations is generated, each iteration consisting of a possible transition from the current configuration to a neighbouring configuration. If this neighbouring configuration has a lower cost, the current configuration is replaced by this neighbour, otherwise another neighbour is selected and compared for its cost value. The algorithm terminates when a configuration is obtained whose cost is no worse than any of its neighbours. There are several disadvantages when using this approach. First, it may terminate in a local optimum and there is generally no information as to the amount by which this local optimum deviates from the global optimum. Secondly, the local optimum obtained depends on the initial solution, for which choice there are generally no guidelines available. Thirdly, it is not possible to give an upper bound for the computation time. Tabu search and Simulated annealing attempt to remove some of these limitations. In the following section, a brief comparison of tabu search and simulated annealing is given.

# 3.5.1 Simulated Annealing and Tabu Search

From the survey of the literature, both simulated annealing and tabu search have been used extensively in timetable construction. It was therefore felt appropriate to conduct a simple experiment to compare their performance in generating a feasible timetable for high school situation. The example chosen was one with 30 weekly periods, 12 subjects and 6 year bands. The 30 periods available were shared between the 12 subjects for each year band in a typical way, with up to 5 periods a week, for the main subjects, down to 1 period a week for the minority subjects. The experiment was conducted on the basis that no subject requires double periods and that all subjects should be "well distributed" throughout the week.

To measure the degree of achievement of the objective of a well-distributed timetable, the total number of different subjects assigned on each day for all year bands was used as a scoring function. If all year bands have no repeated subject on any day, the total score would be a maximum of 180. In contrast, if each year band had some subject twice on each day of the week, the score would only be  $5 \times 5 \times 6 = 150$ .

The same starting solution was used by both methods and was generated from a set of 30 random permutations of any 6 out of 12 subjects. In this way the key constraint that no subject occurs more than once in any period was satisfied. Those permutations were then successively modified by replacing excess subjects with shortfall subjects until all the curriculum requirements had been satisfied. This procedure is described in more detail in chapter 4.

The neighbourhood structures for tabu search and simulated annealing are described in detail in chapters 4 and 5 respectively. They basically involve simple

#### **Background Work**

swaps between cells in the same row of the timetable matrix. Likewise, the cooling schedule for simulated annealing and the tabu list for tabu search are also explained in the following chapters, as are the stopping conditions for both methods.

The experiments consisted of a series of starting timetables being generated which are then improved by both simulated annealing and tabu search. The score obtained by each method was noted. If neither method obtained the maximum score of 180, a starting solution was regenerated and the procedure repeated until one of the two methods yielded a score of 180. The number of regenerations  $\int_{k}^{and}$  the total CPU time on a Hewlett Packard mainframe computer to obtain a maximum score by either method were also noted. The results of a set of six runs are shown in Table 3.1.

From the results in Table 3.1, it can be seen that tabu search consistently yields a greater score than simulated annealing by an average of about 18 points. Furthermore, on none the six repetitions did simulated annealing produce the maximum possible score and only marginally improved the initial score by an average of about 4 points.

On every individual iteration, tabu search came within 3 points of the maximum possible score and there was little correlation between the initial score and the final score (r = 0.51). On the evidence of this limited experiment, a procedure based on tabu search was able to produce a feasible timetable in a relatively short time, and one which would have at most 3 instances in the whole timetable where a subject occurred more than once on any day.

To ensure that these results were replicated for different sized timetables, the experiment was repeated for 35 and 40 periods a week, and for up to 20 different

subjects. Generally, the same pattern of results was obtained, although the number of iterations and the c.p.u. time to achieve the maximum possible score increased with the size of the timetable. The major increase however, occurred with the larger number of subjects; an increased to  $\frac{\alpha}{40}$  period week only marginally increased the number of iterations and the c.p.u time.

No.	Of Runs	Initial Score	Max Score	Max Score	CPU Trime
			S.A.	T,S	
1	1	156	160	178	
	2	160	160	179	
	3	154	158	177	
	4	158	160	178	
4	5	156	164	178	
	6	160	162	180	21.98
2	1	158	161	178	
	2	151	155	.178	
	3	154	159	178	
	4	156	163	180	10.70
3	1	158	159	179	
	2	154	155	178	
	3	159	159	180	8.18
4	1	154	159	176	
	2	155	160	178	
	3	152	159	179	
	4	156	157	179	
	6	156	161	180	16.84
5	1	159	161	178	
	2	159	167	180	5.40
6	1	158	162	178	
	2	152	162	177	
	3	156	158	179	
	4	158	163	180	11.59

# TABLE 3.1: The score of an objective function based on equal subject distribution and the CPU time taken to arrive at a global maximum.

#### **Background Work**

#### Chapter 3

On the evidence of these results, was concluded that tabu search was a more effective procedure than simulated annealing for generating a feasible timetable If medium constraints that satisfied the hard constraints are included, in particular the need for certain double periods, it transpired that tabu search was generally able to satisfy some, but not all, of the double period requirements. It seemed plausible that simulated annealing might be capable of producing worthwhile improvements in the double period allocation when applied to an existing solution which had been obtained by the use of tabu search.

Accordingly, a further experiment was carried out with 2 subjects requiring double periods, each subject needing 3 periods which means that there are 12 double periods allocated over the 6 year groups. Simulated annealing does generally increase the allocation of double periods, typically by between 1 and 3 allocations, which is significant improvement. The quality of the end result is however very much dependent on the starting timetable, that is the initial timetable that was generated using tabu search, which often gives a very good solution. In that case, simulated annealing is looking for further improvement, which is not always possible. The overall process of generating a final timetable is reasonably quick, which allows for a timetable to be regenerated if the result is found not to be acceptable.

## 3.5.3 Rule-Based Expert Systems

Perhaps one of the key issues in understanding and solving a complex real life problem is the ability to articulate at least a partial explanation of the way in which to solve the problem. This explanation is often in the form of a set of rules that describe what to do under different circumstances when tackling problems in some

#### Background Work

well-defined domain. Expert systems are primarily knowledge based in that they tackle complex problems normally solved by human experts, often attaining levels of performance comparable with their human counterparts. The area of timetabling where expert systems are likely to be most useful is in satisfying the soft constraints. These are the constraints that are desirable, but not essential, and which tend to vary from school to school. These constraints may be stored as a knowledge base and can easily be accessed and modified according to what is required, and the satisfaction of each constraint is achieved by constructing an appropriate heuristic. The firing of each of the heuristics is controlled by a rule based program which will be described in Chapter 6.

Each of the approaches described above have their strengths and weakness but they could be combined to produce a complete system that should be capable of producing a good timetable.

# 3.6 The Proposed System

The system is divided into several different systems and modules as shown Figure 3.1 There are three main sub-systems, namely the Input System, the Solution System and Output System, each of which contains several different modules. In the following sections a brief description of each sub-system is given. A detailed description of the implementation of the system is given in Chapter 7.

# 3.6.1 Input Sub-system

This sub-system consists of two main modules; the input and modify modules.

Background Work





# 3.6.1.1 Input Module

The input module reads in the school and curriculum data and prepares this for other sub-systems. Information obtained in this part of the system provides a basis for the whole process of constructing a timetable. The timetabler, after consultation with the head teacher and heads of department, will have a list of subjects to be taught, the number of classes, the number of teachers, the number

#### **Background Work**

#### Chapter 3

of available rooms, which subjects require double periods etc. All this information must be in the possession of a timetabler before entering the assignment stage of the timetabling process. The input module provides the above data along with the basic school data such as the number of year bands, the number of periods per day, the number of days per week and the number and timing of any common periods.

### 3.6.1.2 The Modifying Module

This module allows the user to modify the currently entered data or exit from the data input sub-system. The idea of this module is to allow the timetabler to make any modifications to the basic input data that may be necessary, either before or after a timetable has been generated.

# **3.6.2** Solution Sub-system

A timetable is represented by a two way table with rows representing the year bands and the columns representing the periods in a week. The elements of the table are the subjects.

# 3.6.2.1 Initial Solution Module

A basic feasible timetable is created so that no subject appears more than once in any column (period). This is because all staff in each department will be occupied in teaching the various forms in each year band. This hard constraint is maintained throughout the whole process of construction of the timetable. The process of creating the correct number of periods for each subject as specified in the curriculum matrix must now be undertaken and a separate routine is created to

### Background Work

achieve this. At this point, the generated timetable is basically feasible but does not necessarily satisfy every requirement.

# 3.6.2.2 Intermediate Solution Module

The intermediate solution module provides a solution that will improve the initial solution. In this module techniques of tabu search and simulated annealing are used. An efficient heuristic based on these two metaheuristic approaches is developed to satisfy the common period and the double period requirements. A major feature of this thesis centres on the application of these two techniques and this will be described in detail in Chapters 4 and 5.

#### **3.6.3** Rule-Based Expert Sub-system

The rule-based expert system module is divided into three sub-modules, the rule the building module, the rule-based solution module and/rule evaluation module.

## **3.6.3.1** Rule Building Module

This section will generate interactive questions to the user from which a data base is generated. These data are the information on the soft requirements which will be coded in the form of rules. A series of rules is then built based  $\int_{L}^{COM}$  the entered data, and the firing of each rule is programmed separately. A full description of this section will be given in Chapter 6 and a detailed example of the use of the system will be given in Chapter 7.

# 3.6.3.2 Rule-Based Solution Module

A heuristic model is built for each of the rules given in the above module. For each rule, a heuristic is designed to satisfy the requirements and for each combination of two rules another heuristic is developed. Based on these heuristics a series of rules is fired according to the requirement specified by the user. In all, five rules were defined to satisfy most of the soft constraints likely to be encountered in practice.

# 3.6.3.3 Rule Evaluation Module

For each of the rules fired, the result is tested for its satisfaction. The degree of satisfaction is measured by the number of rule violations, with zero as the best solution. The rule function will determine which of the rule heuristics to fire at any stage to satisfy individual constraints.

#### 3.6.4 Output Sub-system

This output sub-system is divided into two modules, the manual operation module the and/graphic display module.

### **3.6.4.1** Manual Operation Module

The manual operation module gives the user the ability to make changes in assignments (e.g. removing a subject assignment from one timeslot and inserting it into another timeslot). This module is divided into two main components, the edit option and the view option. The edit option is further divided into column swap and cell shift. The view option allows the user to check the timetable day by day

before making any alterations. The procedure for reading and saving files is included in this module so that the user can update and store satisfactory results.

### 3.6.4.2 Graphic Display Module

Once the user is satisfied with the timetable, an option is given to the user to see the final timetable as a graphic display. Three options are provided, namely a display of year groups versus the periods for each day of the week, the days versus the periods for each year group and the subject distributions throughout the week. The user can then proceed to print a hard copy of the timetable.

# 3.7 System Testing

Most of the modules are tested as the system is developed. Finally these modules will be integrated and the whole system will be tested. The testing procedure involves the demonstration of the complete system and hands-on use by the end user, mainly to determine issues like user-friendliness and its ability to solve real problems. Details of the application of the system to three case study schools are given in Chapter 7.

# 3.8 Summary

This chapter represents the main guide to the structure of this thesis. It explains the directions of the research and methods deployed to produce a working and acceptable timetabling system.

The proposed solution approaches the problem by dividing it into linked stages. This division is quite natural to timetabling and it allows for the initial scheduling to be done in such a way that it satisfies the hard constraints and disregards

#### **Background Work**

temporarily the medium and soft constraints. These further constraints (such as common periods and double periods) which are ignored at this stage do not harm the solution process in any way; rather it simplifies the whole process since it concentrates first on producing a basic feasible timetable that satisfies the hard constraints. The medium and soft constraints are considered in subsequent stages. Initial study shows that tabu search is more capable of producing a better feasible solution than simulated annealing.

The second stage of the development involves the assignment of medium constraints such as double period requirements, and fixing common periods to a specified timeslot. At the same time, the procedure also attempts to spread the periods in each subject throughout the week. The third stage of the scheduling problem focuses on the use of the expert system to include the soft requirements. These requirements are set in the form of rules which will direct the search using a heuristic based on tabu search.

Finally, the system is designed to allow the user the opportunity to be directly involved in the final allocation of subjects. The inclusion of an interactive system allows the user to make final manual adjustments and to view and save any timetable which is acceptable. The timetabler may, at the end of the computer session, have several possible timetables to choose from.

Tabu Search

# **CHAPTER 4**

# SOLUTION APPROACH BASED ON TABU SEARCH FOR SOLVING TIMETABLING PROBLEMS

# 4.1 Introduction

Tabu search has been used for solving many combinatorial optimisation problems and its potential for solving timetabling problems has not been fully explored. The aim of this chapter is to investigate the development of a heuristic based on tabu search, for solving high school timetabling problems. Section 4.2 details the development process in building a computerised timetable using tabu search followed by a detailed discussion of the optimisation process for generating a timetable. Finally, an example is given to demonstrate the working of the heuristic approach and its capability to produce an acceptable timetable.

# 4.2 Solution Development

In the context of tabu search related to timetabling problems, a decision has to be made regarding the most appropriate solution procedure to solve this problem. This will involve the choice of the set of feasible solutions, the neighbourhood structure, the evaluation (objective) function, and the way in which starting solutions are generated. All these decisions are related, but in this case finding an initial feasible solution and eventually finding an acceptable solution proved to be the most challenging.

### Tabu Search

For the problem under consideration, the primary requirement of a feasible solution is that each year band is assigned a subject for each period, with no periods having the same subject allocated to two or more year bands. This is because there are only enough teachers in any subject to teach all the classes in one year band. Secondary requirements, for example that the occurrence of repeated subjects on any day should be minimised, could also be considered. Evaluation functions may be defined to satisfy both primary and secondary requirements if necessary.

When defining a feasible solution for tabu search, clashes within periods will cause a major violation of the timetable conditions. Thus an initial solution is an allocation of a subject to each period with no other year band in that period being assigned the same subject. Usually the teacher and room constraints have been solved in advance of constructing a timetable. The person who is responsible for the timetable must know which teachers will teach each subject and in which classrooms. It is a common practice in almost all high schools in Leicestershire to decide in advance for all subjects which teacher is to teach each class within a year band. As a complete year band receives the same subject in any one period, the allocation of teachers to rooms does not usually present a problem.

In the light of the above, an initial solution can be generated by taking each period in turn and allocating a subject randomly to each year band from those subjects which have not already been allocated within that period. The most obvious definition of a neighbourhood is the set of solutions obtainable by replacing any subject in any period by some other subject which has not already been allocated within that period. Finally, the objective function can be expressed in terms of the extent to which the curriculum requirement for all year bands is violated. The implementation of the procedure describe in the previous sections can be divided into the following three main stages.

Tabu Search



From the flow chart given in Figure 4.1, each of the stages will be described in the following sections.

FIGURE 4.1 Stages in Solution Development

The starting solution achieved in Stage 1 will produce a timetable which has no subject taught more than once during any period. This will not be a workable timetable, however, as curriculum requirements have not been considered. Stage 2 will turn the Stage 1 timetable into a theoretically feasible timetable by satisfying all the curriculum requirements (hard constraints). Finally, Stage 3 will provide a more realistic timetable from a student perspective. The completion of Stage 3 will provide a timetable that satisfies all hard constraints and to some degree the medium constraints. The particular medium constraint considered at this stage is the desire to spread the various classes within each subject throughout the school week. The solution may need further adjustment to incorporate other special

#### Tabu Search

requirements such as allocating space for part-time teachers and other requirements which will vary from school to school. This will be dealt with in detail in Chapters 6 and 7.

# 4.2.1 Basic Data Input

Stage 0 consists of basic data that is essential to the generation of a feasible timetable. This is sometimes known as the curriculum data, which in essence defines the academic policy of the school. The information in this basic data concerns the number of subjects taught to each year group, which typically ranges between 9 and 15. This number varies depending on how the timetabler at each particular school wishes to organise the timetable. Most schools divide the subjects into departments but may combine two or more subjects into one department, for example the Department of English may be responsible for both English and Drama, but these are effectively timetabled as one subject. The data takes the form of a two way table of year bands by subjects, with the elements being the number of periods any year band must be taught a particular subject. The number of periods per day, the number of days in a week, the number of year bands is also defined as follows.

- \* the number of days per week (weekly cycle = D)
- \* the number of year bands (year bands = C)
- \* the number of subjects (available subjects = S)
- \* the number of periods (daily periods = P)
- the curriculum requirement matrix A. (the element aij is the number of periods required for year band i and subject j)
- \* the number of periods per week (Total Periods =  $T = D \cdot P$ ).
Tabu Search

A curriculum requirement matrix A consists of elements  $a_{ij}$  where i=1,2,...,C; j=1,2,...S. It is common in high schools that pupils are not allowed any free periods, therefore

**5**  

$$\sum a_{ij} = T$$
 (i = 1,...,C)  
j=1

so that the total number of periods for all subjects must equal the total periods in a week. It is assumed that violation of this constraint will result in a timetable that can be considered infeasible.

# 4.2.2 Starting Solution

Stage 1 creates an initial solution in the form of a matrix R with subjects as its elements. These subjects will be represented in the form of integers 1,2,3...,S for example Tutorial as 1, English as 2 and Mathematics as 3 etc. Considering each year band in turn, a subject is generated at random for each period. In any period, those subjects already assigned to previous year bands are not allowed. An example of a timetable matrix for 6 year bands is given in Table 4.1, where each set of five columns represents a five period daily timetable for one day in a five day weekly cycle.

#### Chapter 4

							P	erio	ds								
Band		M	ond	av_				T	uesd	ay				We	dnes	day	
1	2	7	7	4	1		8	5	2	9	2		8	4	1	9	9
2	4	6	2	5	2		5	3	6	1	4		5	9	8	8	3
3	3	3	5	1	5		6	7	8	6	8		6	7	7	3	2
4	1	5	8	3	3		9	2	1	4	7		3	8	4	5	6
5	6	2	3	8	7		3	4	9	5	1		4	2	3	6	5
6	7	4	6	6	9	Π	4	6	3	7	5	Π	2	3	5	4	4
Band		Th	urs	iav				F	'rida	IV.							
1	7	4	2	3	7		4	3	3	9	5						
2	2	3	4	5	3		9	6	2	5	7						
3	6	5	8	7	2	Π	5	2	8	8	4	Π					
4	4	9	6	2	6	Π	1	7	4	2	3	Π					
5	1	7	3	9	4	Π	2	4	7	6	6						
6	5	2	9	1	8		8	5	5	4	2						

TABLE	4.1:	Ma	trix	R
-------	------	----	------	---

# 4.2.3 Producing a Feasible Timetable

Certain periods, such as a common tutorial period (where all year bands have the same lesson) or a period set for all years to meet for an assembly, must be allocated to defined slots. These requirements are satisfied by preassigning them in one or more of the required columns in **R**. In this case, the common period is considered to be a subject and it is assigned *as subject one*. This subject, once assigned, must not be interfered with subsequently. For example, if a common tutorial (subject 1) is required in the first period on day 1 and in the third period on day 3, then  $r_{ik} = 1$  (i=1,...,6; k=1,13). Columns 1 and 13 in **R** are then locked to prevent subsequent alteration.

Each column in R can be considered as a random permutation of C different integers in the range 1 to S. A two-way table of T such permutations gives a *starting timetable* in terms of S different subjects. The frequency of each subject in each year band is determined and placed in a generated curriculum matrix A'. As already noted, the timetable generated at this stage will not necessarily satisfy the curriculum requirements unless

(all i, j)

where  $a'_{ij}$  denotes the number of occurrences of subject j in row i of the generated timetable matrix R. An example of the generated curriculum matrix for the data in Table 4.1 is given in Table 4.2

Subjects

Bands	1	2	3	4	5	6	7	8	_9
1	2	4	3	4	2	0	4	2	4
2	1	4	4	3	5	3	1_	2	2
3	1	3_	3	1	4	4	4	5	0
4	3	3	4	4	2	3	2	2	2
. 5	2	3	4	4	2	4	3	1	2
6	1	3	2	5	5	3	2	2	2

## TABLE 4.2 Generated Curriculum Matrix A'

# 4.2.4 Deviance matrix

Stage 2 aims to create a timetable which satisfies the curriculum requirements, by considering the deviance matrix

$$\mathbf{D} = \mathbf{A'} - \mathbf{A}$$

# Chapter 4

Matrix D indicates the excess or shortfall in the number of periods of subject j timetabled for year band i.

				Danl	cc.s				
Bands	1	2_	3	4	5	6	7	8	9
1	1	3	3	4	3	2	2	3	4
2	1	4	4	3	3	2	3	2	3
3	1	4	4	4	3	3	3	2	1
4	1	4	4	4	3	3	3	2	1
5	1	4	4	4	4	3	3	1	1
6	1	3	4	4	4	4	2	2	1

Subjects

# TABLE 4.3 Curriculum Matrix A

If the curriculum requirements matrix A is as shown in Table 4.3, then the difference between Table 4.2 and Table 4.3 gives the deviance matrix in Table 4.4.

	1	2	3	4	5	6	7	_8_	9
1	1	1	0	0	-1	-2	2	1	0
2	0	0	0	-1	2	0	-1	0	0
3	0	-1	-1	-2	1	2	1	3	-3
4	2	-1	0	0	-1	0	-1	0	1
5	1	-1	0	0	-2	1	0	0	1
6	0	0	-2	1		-1	0	0	1

TABLE 4.4 Deviance Matrix D

Clearly the matrix **D** will be stochastic with

S  

$$\sum d_{ij} = 0$$
 (i = 1,...., C)  
j=1

A common move mechanism for permutation problems is *replacement*, which consists of replacing the contents of an element in the permutation by some other subject. For each year band i, any subject with excess periods ( $d_{ij} > 0$ ) must be replaced by another subject. The replacement can be any subject which has a shortfall in row i of the deviance matrix ( $d_{ij}<0$ ) and which does not appear already in column j. In other words, those subjects with  $d_{ij}\geq 0$  and which are already present in period j are tabu. For example, subject 1 has an excess of 1 for year band 1 ( $d_{11} = 1$ ) and we wish to replace (say)  $r_{15} = 1$  by some other subject. For such a replacement, subjects 2,3,5,7 and 9 are already present in period 5, and subjects 2,3,4,7 and 9 are already exactly satisfied or in excess for year band 1. The tabu list therefore comprises subjects 2,3,4,5,7 and 9. If  $r_{15}$  becomes (say) 6, then this produces deviance function values  $d_{11} = 0$  and  $d_{16} = -1$ .

By a series of such *replacements*, the starting timetable R will be transformed into a feasible timetable which satisfies the curriculum requirements. In applying tabu search, the neighbourhood set will be defined by those solutions which can be produced by simply replacing some element (subject) in R which has a positive count in the corresponding row of D by some other subject. The tabu list will define which swaps are not legitimate and will be made up of all subjects which have a positive or zero count in D, or which are already present in the same column of R (i.e. are being taught to some other band during the period under consideration). While legitimate replacements are available, the subject with the largest positive count in the deviance matrix is chosen, and the first occurrence of that subject where a replacement is possible is replaced by a subject chosen at random from those available to replace it.

Ultimately, however, as the timetable approaches feasibility, it is likely that no legitimate replacements will be possible before all the  $d_{ij}$  elements have become

#### Tabu Search

zero. If and when this occurs, the definition of the tabu list changes slightly and now no longer includes those subjects with a zero count in the appropriate row of the deviance matrix. The effect of this is to permit swaps with subjects whose curriculum requirement is exactly satisfied and thereby replace one excess subject by another. The procedure then reverts to the original definition of the tabu list and proceeds as before until no legitimate swaps are available, or until the deviance matrix has become zero.

At the end of this process, more often than not a feasible timetable is generated. This matrix will be used as an initial feasible solution in solving the next stage of the timetabling process. In the unlikely event that a feasible timetable cannot be generated in this way, the procedure returns to stage 1 and a new set of random permutations is created and the process starts all over again.

# 4.3 Satisfying Medium Requirements

Medium requirements can be divided into two main areas, namely satisfying spread and satisfying double periods.

# 4.3.1 Satisfying Spread

Stage 2 above dealt with the *hard* constraints, namely satisfying the curriculum requirements and at the same time maintaining no clashes in any period throughout the week. The medium or highly desirable constraints are now considered, for example, that the subjects should be evenly spread throughout the week, and a class should not have the same subject in two consecutive periods or perhaps appearing more than once on any day unless they are deliberately planned a multiple periods. Examination of the timetables produced by Stage 2 shows that there are frequent

repetitions of subjects within the same day. Indeed, the major subjects which have 4 or 5 periods devoted to them could have as many as 4 periods in the same day, which is highly undesirable. In some situations, there may be a requirement for double periods for some subjects such as Design and Science. This situation is discussed in the next sub-section. In some schools, all periods were explicitly planned as single periods and it was felt desirable to have as few occasions as possible when the same subject was taught twice (or more) on the same day. A timetable is said to be well spread if there is a 'large' number of different subjects on each day. An obvious objective function is therefore to maximise the total number of different subjects taught in any day summed over all year bands and all days.

In the case where no multiple period assignment is required, then the total number of cells in the R matrix can be used as the target score, otherwise the total minus the number of multiple period requirements is set as the target score. If the score obtained is less than the target score then the procedure is repeated until a better solution is achieved. For a timetable with D days, C year bands and P daily periods, an evenly spread timetable will have a maximum possible value for the objective function of CxDxP. Using the example given above the maximum possible value of the objective function is 150 (C=6, D = 5 and P = 5). Imposing double period requirements into the table will necessarily reduce this number. A procedure to handle double periods is proposed in the next section.

# 4.3.2 Satisfying Double Periods

Satisfying double periods means having to allocate a subject to consecutive periods. In a limited way this goes against the principle of increasing spread as described in the above section. An efficient heuristic with the ability to satisfy double period

#### Chapter 4

allocations while at the same time maximising spread must be created. Combining these two features, namely spread and double period requirements, appears to be a reasonable strategy. By modifying the objective function to include a double period requirement it should be possible to not only create the necessary double periods but also prevent having a timetable with additional subject repetitions on the same day. Here the two elements are simply added to give the score function

$$C \quad D$$
  
Score Function =  $\sum_{i=1}^{N} \sum_{k=1}^{N_{ik}} (N_{ik} + Q_{ik})$ 

where,

6

Nik is the number of different subjects taught to band i on day k.

Q<sub>ik</sub> = 1 if a subject requiring double periods for year band i is assigned to successive periods on day k.

= 0 Otherwise

The procedure begins with a compromise between the need to have non-repeated subjects throughout the week and allocating subjects which require consecutive periods. Any allocation of double periods with one period on either side of lunch break is considered to be invalid. With multiple requirements which may be conflicting, a decision must be made about which are more important and therefore carry more weight. Satisfying double period requirements is usually more desirable and therefore will override the equal spread requirements. The preference for double periods can be expressed by weighting that component in the score function given earlier, i.e.

Chapter 4

Score Function = 
$$\sum_{i=1}^{C} \sum_{k=1}^{D} (N_{ik} + WQ_{ik})$$

- N<sub>ik</sub> is the number of different subjects taught to band i on day k.
- W is an appropriate weighting factor.

Q<sub>ik</sub> = 1 if a subject requiring double periods for year band 
$$i$$
 is  
assigned to successive periods on day k.

= 0 Otherwise

The weight can be increased or decreased depending on how strong the double period requirement is. Maximising this function will maximise the required number of double period allocations and at the same time maintain a good subject distribution. This is carried out using a heuristic procedure based on tabu search.

The tabu search procedure to achieve this requirement begins by calculating the number of double period requirements for each year band. These requirements are placed in a matrix D' where the value of  $d'_{ij}$ , a non-negative integer, is given as

$$d'_{ij} = \begin{cases} 0.5[a_{ij}] \text{ for subjects requiring double period} \\ 0 \text{ Otherwise} \end{cases}$$

where  $a_{ij}$  is the required number of teaching periods for class i and subject j. For the process of allocating subjects in **R** so that subjects with double period requirements are placed consecutively, a heuristic procedure is used as described in the following general algorithm.

Step 0: Generate a feasible timetable R satisfying curriculum requirement. Set iteration number equals 0.

Chapter 4

Calculate the score function

Step 1 :	For each year band: Search for subject j with $D' > 0$ .
	If found, mark the column as ColOne and day as DayOne.
	Check ColOne+1 to see if $R(i, ColOne+1) = j$ .
	If Yes repeat Step 1 for next j.
	If Not put elements in ColOne+1 into tabu list.
	Goto Step 2.
Step 2 :	Randomly generate a day and call it DayTwo.
	If DayTwo = DayOne repeat step 2.
	If Not, locate subject j.
	Goto step 3
Step 3 :	Is subject in R(i, ColTwo) in tabu list?
	If Yes Goto step 2.
	If Not, do swap operation.
	Calculate score function. If improved, accept swap
	If Not, Goto step 2.
	Goto step 4.
Step 4 :	Block subjects already satisfying double period requirements.
	Remove any clash in ColTwo (Call clash removal routine)
	Goto step 5.
Step 5 :	Modify double period requirement matrix D'
	If $d'_{ij} = 0$ (all i,j) then stop.
	Check Iteration number. If iteration $> k$ then stop
	Goto step 1.

If the clash removal routine is required in step 4, the procedure puts the subjects in ColTwo into a new tabu list and searches for another column within R which does not contain the clashed subject in Coltwo and where the subject in row i is not the

#### Chapter 4

It

new tabu list nor is blocked. such a column can be found, the row elements are swapped. If such a column cannot be found, the previously created double period is undone and the procedure returns to step 2.

The updating of the tabu structure is a simple operation. The attributes of swap are stored in the structural variable Tab(n), where n is the size of the tabu list. The program uses a dynamic tabu list mechanism (maximum is set at 17) which is used for diversification and for decreasing the probability of cycling. The procedure checks for the satisfaction of double period requirements and will repeat the procedure if the requirements are not met. The stopping criterion will be the number of iterations specified in advance. The number of iterations required is normally not more than 20.

# 4.4 **Results and Discussion**

The tabu search procedure was tested on several sets of hypothetical data which are similar to the real timetabling situation in high schools. To determine the general effectiveness in terms of producing a complete timetable, and also the time taken to generate a timetable, the data sets used contained different numbers of subjects and different numbers of periods in a different weekly cycle.

Starting from a random permutation, the results of the experiment show that a greedy heuristic that selects the first improving swap at each step is highly effective in finding a solution and at maintaining no clashes within a period. This approach has been demonstrated to be able to solve problems ranging from 4 to 10 year bands, from 7 to 15 subjects and for between 25 and 40 periods a week: In practice, a typical number of year bands, subjects and weekly periods are 6, 12 and

# Chapter 4

30 respectively. The experiment always starts with a solution and a realistic solution is obtained, regardless of the problem size and the initial random solution.

An experiment to determine the average CPU-time (on a Hewlett-Packard Mainframe computer) in generating a timetable for variable weekly periods based on 10-20 subjects and 6 year bands was carried out. The time taken to generate a realistic timetable for 30, 35 and 40 weekly periods is shown in Figure 4.2.



FIGURE 4.2: Effect of number of subjects and number of weekly periods on CPU time

This indicates that an increase in the complexity of the problem, in terms of the number of subjects and the number of periods in a weekly cycle, does increase the CPU-time. Overall, however, the process is relatively quick and can usually produce a feasible timetable in less than 0.3 seconds CPU-time.

This limited experimentation with different numbers of subjects and periods also shows that the CPU-time to produce a feasible solution increases more or less

Tabu Search

linearly with the number of subjects. In contrast, there is some evidence that the increase from a 7 to 8 period day takes proportionately longer to timetable than the increase from 6 to 7 periods per day.

The primary aim of producing a timetabling system is to assist timetablers in high schools to produce a timetable for their school that requires as little modification as possible, and in reasonable time. The approach using tabu search is capable of producing a working timetable in a short time. Nevertheless, there are certain occasions where tabu search could not fully satisfy the requirements of double periods. This is the main area that takes a lot of the execution time as more often than not the procedure has to generate a new initial timetable before finally arriving The process of allocating the double periods is at an acceptable solution. determined by a weighting function which gives priority to subjects with double period requirements, so that when such subjects are placed side by side, a bonus score is given. If too heavy a weight is given to double period requirements then it will lead to a high number of repeated subjects within a single day. After experimentation, a value of 15 was chosen as a fixed weight which satisfied almost every double period requirement. This seems to give a reasonable compromise between satisfying most (if not all) double period requirements and achieving a reasonable spread of subjects through the week.

## 4.5 Summary

This chapter demonstrates and discusses the implementation issues that relate to the use of tabu search for finding a solution to a high school timetabling problem. This chapter also shows how to incorporate tabu search strategies and examines the effect of such strategies on the execution time.

#### Chapter 4

Results shows that tabu search is usually capable of directing the search to a satisfactory solution. Its execution time is considered to be fast and well suited to run in a PC environment. In several trials, the heuristic model has demonstrated its capability to generate a feasible timetable in a short time. This experiment was carried out on a Hewlett-Packard Mainframe computer.

The heuristic model based on tabu search seems to capture much of the complexity of typical high school timetabling problems, and it allows the use of weight parameters into the objective function to achieve the assignment of double periods. Although the model does not always satisfy every requirement, it did solve the major part of the problem from which further modifications could be done manually. The results indicate that subjects can be allocated without any clashes, and the double period requirements can usually be accommodated provided that there are not many of them.

As difficulties sometimes arise in producing a timetable that satisfies every double period requirement, further work could usefully explore this problem using another method. One possible extension to this model could be to employ a heuristic method known as simulated annealing. A discussion of the adaptation of the simulated annealing technique to generate a feasible timetable and improve the double period allocation is presented in the next chapter.

# **CHAPTER 5**

# FURTHER REFINEMENT OF THE TIMETABLE USING SIMULATED ANNEALING

## 5.1 Introduction

In the previous chapter, a heuristic model based on tabu search proves to be capable of producing a *good* solution to high school timetabling problems within a reasonably short time. Nevertheless, these solutions can still be improved especially in the area of satisfying multiple period requirements. There has been considerable interest in the use of simulated annealing for obtaining *good* solutions to a wide range of large combinatorial problems. This chapter deals with the development of an efficient heuristic model based on simulated annealing for refining the previously generated timetable. First, it deals with the improvement of subject distribution in timetables based on tabu search. Second, it looks at improving double period assignments which were initially produced using tabu search. The search process uses swap and replacement operations between periods and between cells in the timetable matrix.

Simulated annealing has proved successful as a randomised search technique for many kinds of optimisation problem. It often gives good solutions, even when more conventional optimisation methods get trapped in a local optimum. Recently there has been considerable interest in the use of simulated annealing as a technique which reduces the likelihood of local convergence. One of the areas where simulated annealing has been used is in solving timetabling problems. Johnson (1990) used simulated annealing for solving an examination timetabling

#### Simulated Annealing

problem, Hertz and de Werra (1987) described its use for edge-weighted-colouring problems which were then used in solving a school timetabling problem, Eglese and Rand (1987) applied it to a conference scheduling problem in which the objective is to minimise disappointment and Dowsland (1990) used simulated annealing to solve a timetabling problem in which clashes are inevitable.

This chapter commences with a description of the timetabling annealing scheme designed to solve timetabling problems. Its implementation requires some definitions, and these are given in Section 5.2. Section 5.3 examines how the solution generated in the last chapter can be improved regarding the assignment of double periods and the subject distribution. Finally, a discussion of the results is given in Section 5.4.

# 5.2 The Timetabling Annealing Scheme

This section explains the use of simulated annealing to obtain an improved solution to a high school timetabling problem. The aim is to remove as far as possible any repetition of subject allocations in a single day for each year band. Before proceeding with the description of the scheme, a list of definitions is given in the next section in order that the use of simulated annealing techniques for solving timetabling problems can be understood.

# **5.2.1** Definitions

An *element* is a particular combination of year group, period and day where a particular subject is taught.

#### Simulated Annealing

An objective function for evaluating a given timetable is a measure of the quality of the solution. It also acts as a measure of achievement of the specified requirements.

A score function calculates the number of non repeated subjects in a day plus the number of allocations of double periods. The assignment of double periods is controlled by the weight (or bonus) which encourages those subjects with double periods to be set next to each other. Hence the optimisation is one of maximising the value of the score function.

The cooling rate is a measure of how quickly the temperature is decreased.

A cooling schedule for a given optimisation problem is characterised by a fixed initial temperature  $(T_0)$ , and a rate of decrement over a defined number of steps such that  $T_n = f(T_{n-1})$  where f(.) is a monotonically decreasing function.

Other definitions such as the timetable matrix, curriculum requirements, subjects, lessons and topics will remain the same as defined in Chapter 4, Section 4.2.

#### 5.2.2 The objective function

The objective function involves three components, namely a score function for clashes within a period, a score function for double period requirements and a score function for subject distribution.

The first component is to try and prevent clashes from occurring within any period. This is necessary because there is a possibility that the swapping of elements from cell to cell can cause clashes to occur. The second component in the objective

**75**;

#### Simulated Annealing

function is to assign subjects requiring double periods. Satisfying this requirement may also cause other subjects to appear more than once in other days, which is generally undesirable. In order to overcome this problem, another part of the objective function must control the subject distribution. As before, this requirement is dealt with in the objective function by introducing a third component which will try to prevent the appearance of any subject more than once in each day by maximising the number of different subjects taught to each year band on any day. The number of repetitions in a day for the whole week is measured by the difference between the total number of non repeated subjects and the number of repeated subjects in a day.

The relative importance of different components in the objective function is controlled by a series of weights. One such weight is that for the double period allocation, which measures the importance of a particular subject being assigned as a double period. A large weight means allocation of a subject as two consecutive periods is highly desired; a small weight not favour such an allocation. The value of the weight varies depending on the degree of desirability.

Ideally, the weight should be a dynamic one which starts with an initial value and is subsequently increased to satisfy the need for double periods. For example, the weight may be initially set as 3, but if the number of double periods assigned is far less than the desired number, then the weight is automatically increased to 4 in the next iteration. The weight will continue to increase until it reaches a defined maximum weight.

Hence, the overall objective function to be maximised is as follows

**Simulated Annealing** 

$$F = \sum_{k=1}^{D} \left[ X \sum_{j=1}^{P} G_{ik} + W \sum_{i=1}^{C} Q_{ik} + \sum_{i=1}^{P} N_{ik} \right]$$

where

 $N_{ik}$  is the number of different subjects taught to year band i on day k.

 $G_{ij} = 1$  if all subjects assigned in period j on day k are different.

= 0 Otherwise.

 $Q_{ik} = 1$  if a subject requiring double periods for year band i is assigned to successive periods on day k.

= 0 Otherwise

- W is a weighting factor to reflect the relative importance of double periods.
- X is a weighting factor to reflect the relative importance of a no clash timetable.

One difficulty in setting up this objective function is to strike a balance between all three requirements. Even though the first part of the objective function is an essential requirement, such that any violation is intolerable, imposing too high a weight may cause stagnation in the search for a better solution. A similar situation applies to the second requirement. So, the role of simulated annealing is to search for a satisfactory timetable in terms of having the highest value of the objective function. This should produce a no clash timetable that satisfies most, if not all, of the double period requirements with a good distribution.

## Simulated Annealing

# Chapter 5

# 5.2.3 Initial allocation

In order that the simulated annealing phase has a solution upon which to try and improve, a child program is executed. This program produces a basic feasible timetable using tabu search to give a reasonably *good* feasible solution which satisfies some of the double period requirements. There is still some room for improvement as the initial timetable usually does not satisfy all double period requirements. The result generated is stored as an initial timetable matrix R. From now on, whenever the initial feasible timetable is referred to, it is the timetable generated using this child program as described in the previous chapter. A sample of the initial allocation is given in Chapter 4, Table 4.1.

# 5.3 Adaptation of simulated annealing

In this application, the atoms are replaced by the elements of matrix R. The system energy is replaced by the score value of each table. Score is used to reflect the "quality" of the timetable, just as the system energy reflects the quality of a substance being annealed. The temperature is used to control the probability of an increase in score analogous to the temperature of a physical substance. Since this annealing algorithm relies on a random set of permutations it cannot guarantee that the true maximum score value can actually be found or that two annealing runs will yield the same solution.

The procedure for improving spread and double periods is tackled by two operations, namely by swapping two periods, or by replacing individual elements. Before discussing the actual swapping and replacing operations, several features of the procedure will be described.

#### Simulated Annealing

## Chapter 5

# 5.3.1 Cooling Schedule

There are several parameters described in Van Laarhoven and Aarts (1985) which govern the simulated annealing algorithm. In this section, a simpler approach is followed, which is similar to the approach used by Johnson (1990) for solving an examination timetabling problem and Eglese and Rand (1987) for a conference scheduling problem. This approach is highly dependent on the way the control parameter decreases, and a procedure is needed to govern the operation of the run time parameters. This set of parameters constitutes what is commonly known as a *cooling schedule*. A cooling schedule specifies the following features to govern the convergence procedure:

- an initial value of the control parameter T<sub>0</sub>
- a decrement function to reduce the value of the control parameter
- a final value of the control parameter
- a termination of run criterion

Since simulated annealing is based on randomised search, there is no guarantee that it can find a global optimum to the problem. However, if there is a chance of arriving at a global optimum, then with a careful decrease of the control parameter it will hopefully reach that global optimum. A brief summary of the cooling schedule is now given.

#### 5.3.2 Initial Values of the Control Parameter

The basic assumption in setting the initial value of the control parameter  $T_0$ , the initial temperature, is that it should be sufficiently large. This is to ensure that virtually all transitions will be accepted at this value. Lundy and Mees (1986) suggest that  $T_0 \gg U$ , where U is an upper limit on the change in the objective

#### Simulated Annealing

function between neighbouring solutions. Johnson (1990), in solving an examination timetabling problem used a similar approach and successfully obtained satisfactory convergence in reasonable time. This was carried out on a set of trial problems where the true optimum had been determined for all possible permutations. Convergence effectively occurred after about 3000 iterations, after which little change to the score function was observed.

Initial experiment shows that the temperature was reduced by an average of 0.01 per iteration. As the objective function is based on an initial solution where some of the double period requirements have already been assigned, its increments are very small and thus the temperature needed will also be moderately small. A starting temperature of 10 proves satisfactory, but in fact the solution can be achieved at a temperature much lower without reducing the "quality" of the solution. In order to be sure that it will allow virtually all transitions  $t_0$  take place, the initial temperature is set at 10. Too large an initial temperature will only cause the search to continue without actually making any improvement.

## 5.3.3 Decrement of the Control Parameter

The subsequent values of temperature T are controlled by a Lundy and Mees (1986) continuous cooling scheme which is defined by the recurrence relation

$$T_n = T_{n-1} / (1 + bT_{n-1})$$

where the constant  $b \ll T_0$  is chosen so that the temperature falls from the initial value  $T_0$  to a final temperature  $T_f$  in M iterations. This implies that

$$\mathbf{b} = (\mathbf{T}_0 - \mathbf{T}_f) / \mathbf{M} \mathbf{T}_0 \mathbf{T}_f.$$

#### Simulated Annealing

# 5.3.4 Termination of the Run

The run is stopped if the score function has reached its maximum possible value (i.e. upper bound) namely the difference between the total number of non repeated subjects and the number of double period requirements (C\*D\*P - 0.5\*E; where E = total periods for subjects needing double periods). In this case an optimum timetable has been computed and there is no point in continuing the search. The second condition arises if the score function value does not change for a certain number of iterations MR. This is to make sure that it does not continue searching after failing to reach a better solution in a specified number of rejected moves is counted and if this becomes greater than the parameter MR, then the cooling the value is solution was found, and the search continues from this temperature and performs the remainder of the M iterations. The run will also be terminated if it has reached the final temperature.

## 5.3.5 Condition for Accepting A Move

A move, in this case a swap operation, can take place provided the new score function is greater than the original score function. If this is so, update the current score function to the new value and calculate the current temperature T. It is also necessary to record the maximum score function obtained so far, along with the associated timetable. This is to make sure that the chosen timetable corresponds to the best score function obtained overall, which may not be the current situation.

If this condition is not satisfied, a move can still take place provided the calculated probability is greater than a randomly generated value. This is one of the characteristics of simulated annealing which allows the search to go back to a

## Simulated Annealing

# Chapter 5

lower score before going forward to search for a better solution. If the calculated probability is less than then the randomly generated value, this move is rejected and another swap is randomly chosen. There will be a time when the search becomes saturated and the procedure then stops.

# 5.4 Satisfying Spread and Double Periods

Satisfying double periods means that a specified subject must be assigned to consecutive periods. There are two ways that this requirement can be achieved.

- a. by swapping two complete periods (i.e. columns), and
- b. by swapping and shifting cells in the timetable matrix.

To demonstrate this, consider a simple two day timetable as shown in Table 5.1.

	Da	yon	e					Day	two	
Year Band	1	2	3	4	5	1	2	3	4	5
1	2	7	7	4	1	8	5	2	9	2
2	4	6	2	5	2	5	3	6	1	4
3	8	3	5	1	5	6	7	4	6	3
4	1	5	8	3	3	9	2	ł	4	7
5	6	2	9	8	9	3	4	7	5	1
6	7	4	6	9	6	4	6	3	7	5

TABLE 5.1: Two days timetable matrix R

For example, if each year band requires subject 2 to be assigned as double periods then in the first year band, period 2 in *Dayone* may be swapped with either period 3 or period 5 in *Daytwo*. Swapping period 2 in *Dayone* with period 3 in *Daytwo* gives the result shown in Table 5.2. A similar operation can be carried out between periods 4 and 5 in *Dayone* and thereby creating a double period for subject 2 in year band 2. The result is that these swap operations bring subject 2 together in

#### Simulated Annealing

### Chapter 5

both year bands 1 and 2. Once these double periods have been assigned, they will be placed in a list of forbidden moves. Any further operations must not involve these periods (period 1 and 2 in DayOne or period 4 and 5 in DayOne). This swap operation between periods is employed because of the simplicity of the operation in satisfying double period requirements and at the same time avoiding any clashes within periods.

•		Day	one	;		]	Dayt	wo		
Year Band	L	2	3	4	5	1	2	3	4	5
1 (	2	2	12	F	4	8	5	7	9	2
2	4	6(	2	2	25	5	3	6	1	4
3	8	4	5	5	1	6	7	3	6	3
4	1	1	8	3	3	9	2	5	4	7
5	م	7	9	9	8	3	4	2	5	1
6	7	3	6	6	9	4	6	4	7	5

TABLE 5.2: Effect after swap operation to matrix R.

Unrequired Double Period

This procedure has two clear disadvantages. First, by blocking any two periods already assigned as a double period, it will automatically reduce the degree of movement (number of choices) available for satisfying subsequent double period requirements. Secondly, by satisfying double period requirements, it may worsen the subject distribution throughout the week. Thus, it causes an increase in the objective function due to satisfying double period requirements but may cause a reduction in the objective function due to not satisfying the required weekly subject distribution. For example subjects 1 and 3 are allocated twice in Dayone for year band 4, both cases as unrequired double periods. These are not the only examples of multiple subject assignments in Table 5.2. Simulated annealing plays an important role in searching for a balance between the need for double periods and the need to minimise subject repetitions within days.

#### Simulated Annealing

The search to satisfy double periods and minimise subject repetitions in a day using simulated annealing begins with an initial feasible timetable R as a result of the tabu search phase. A period with a subject requiring double periods is identified and checked to see whether it has already been assigned as a double period (i.e. *js* the same subject is next to it). If it does not satisfy double period requirements then the period after it (or before it if it is the last period in a morning or an afternoon) is marked as the *source period*. This period is to be swapped with another period that contains the required subject within that year group. A succession of random days and periods is generated until another one is found that contains the subject required to be assigned as a double period. If found, this period will be marked as the *destination period*. A swap operation may now take place provided the condition for accepting a move is satisfied, otherwise another random day and random period are generated.

There are several disadvantages when swapping the whole period. First, as mentioned earlier, it may cause a poor distribution of subjects throughout the week. Secondly, not all periods can be shifted to a new location without interrupting previous assignments in other year groups. Therefore it is sometimes not practical to simply swap whole periods. Even though it is simple to swap whole periods to satisfy double period requirements, and at the same time prevent clashes, it would be much more flexible if individual cells could be swapped instead of the whole column. But if an individual cell is swapped it may cause clashes within periods which is clearly demonstrated in Table 5.3. For example, consider  $r_{24} = 5$  with element  $r_{25} = 2$  in Table 5.1 to create Table 5.3 which satisfies one of the double period requirements for subject 2 in year band 2 ( $r_{23} = r_{24} = 2$ ).

#### Simulated Annealing

#### Chapter 5

		Dayone Daytv								wo		
Yea	r Band	- 1	2	3	4	5	1	2	3	4	5	
	1	2	7	7	4	1	8	5	2	9	2	
	2	4	6	2	2	3	5	3	6	1	4	
	3	8	3	5	J	5	6	7	8	6	3	
	4	1	5	8	3	3	9	2	1	4	7	
	5	6	2	9	8	9	3	4	7	5	1	
	6	7	4	6	9	6	4	6	3	7	5	

However, this operation has caused a major violation with subject 5 now appearing twice in column 5. This method of swapping between cells can be used but it requires a careful heuristic approach to overcome such clashes while searching for the required double period allocations. In simulated annealing, all moves will be evaluated using the objective function and a large weight (X) is imposed on clashes as described in Section 5.2.2. A large enough will tend to prevent any classes from occurring.

At the end of the run, the best timetable obtained is checked for possible clashes within any period. Usually clashes do not occur when using this procedure as this feature is included in the objective function. If it does occur however, timetable cannot be accepted and the simulated annealing procedure is repeated by starting again from the initial timetable produced by the tabu search phase.

# 5.5 Results and Discussion

An experiment to compare simulated annealing with tabu search as a means of generating an initial timetable was carried out and showed that tabu search can generally produce a timetable with a much better spread than simulated annealing.

Simulated Annealing

However, when applied to creating a timetable with double period requirements, simulated annealing usually managed to improve on the timetable produced initially by tabu search. An experiment was carried out with 2 subjects requiring double periods, each subject having 3 periods in total, which means that there are 12 double periods to be allocated over the 6 year bands. Simulated annealing did always increase the allocation of double periods by 1 and 3 allocations, which is a significant improvement. The quality of the end result is however very much dependent on the starting timetable (that is the initial timetable). In this case, the initial timetable was generated using tabu search, which often gives a very good solution, and simulated annealing is looking for further improvement, which is not always possible.

The process of generating a final timetable is reasonably quick. It typically takes less than 5 minutes to produce a feasible timetable on a 486 PC which can then be modified interactively. This allows for a timetable to be regenerated if the result is found not to be acceptable. As mentioned earlier, some degree of manual adjustment is required in almost every timetable program, so this system also provides such facilities. One advantage of this combined approach using tabu search and simulated annealing is the speed, which allows the timetable to be adjusted almost instantly. The manual scheduling included in this system will be discussed in detail in the following chapter.

# CHAPTER 6

# RULE-BASED EXPERT SYSTEM FOR SOLVING TIMETABLING PROBLEMS

# 6.1 Introduction

The last two chapters have demonstrated the use of tabu search and simulated annealing for solving timetabling problems with hard and medium constraints. As described in the First chapter, however, there are also soft requirements that should be satisfied. Satisfying these requirements is not critical in timetable construction but it would be 'nice' if they could be satisfied. These soft requirements are not common to all high schools and often each school has its own special requirements. Therefore constructing a computer based timetabling system using a conventional programming approach would be very difficult and complicated.

As stated in the literature review, the basic mathematical and OR approaches (such as graph colouring) are useful for understanding and defining the problem, although when applied in real circumstances, they may perform rather inadequately. This is mainly due to the number of variables and constraints involved, which often grows exponentially with the number of year groups, classes and teachers. Even using heuristics to control the combinatorial explosion, this approach often remains rather inefficient in computational time. As demonstrated when using tabu search and simulated annealing, there are also difficulties in defining a suitable cost function to measure the attractiveness of solutions. As an alternative, this chapter considers the use of Artificial Intelligence techniques for finding, not the optimal solution, but a more acceptable one.

#### Rule Based Expert System

The search for a solution to timetabling problems often does not take into account the soft requirements mentioned earlier, but these often determine the quality of a timetable. A major difficulty in constructing a timetable using a computer is in understanding how timetablers solve their problems or more specifically how to articulate an explanation of the way in which they go about solving the problem. This chapter describes the building of a rule based expert system to search for a further improved timetable.

The chapter begins with a definition of an expert system, including the concept of expert system technology and knowledge bases using production rules. It then considers the process of developing a rule base to reflect the various kinds of soft constraints that arise in school timetabling. A full description of the timetabling system using this rule base will be given in the next chapter.

# 6.2 Expert System - A Definition

The key element in solving problems that face humans as experts, is understanding  $\alpha$  complex real life problem and  $\alpha$  articulating an explanation of how to solve the problem. These explanations are often in the form of rules that describe what to do under different circumstances when tackling a problem. A program which automates the application of these rules is known popularly as an *expert system*.

An expert system comprises " computer software that uses knowledge (rules about a particular subject domain) and facts and inference techniques to solve/problem that usually requires the ability of a human expert" (Walter and Neilsen 1988). It can also be described as a computer program that does intelligent analysis of data, which creates a dialogue with the user and can generate further analysis based on

#### **Rule Based Expert System**

the data entered. There are many different definitions of expert systems provided in the literature, but they all emphasis three major features (Kastner & Hong, (1984); Assad & Golden, (1986) and Fordyce *et al* (1987)). These are that expert systems are primarily knowledge based, that they solve complex problems normally solved by human experts; and finally that they attain levels of performance comparable with their human counterparts.

# 6.3 Characteristics of Expert Systems

Figure 6.1 shows a basic expert system structure in relation to the expert, the expert system developer, who is normally referred to as the knowledge engineer, and the end-user. A basic expert system is characterised by four components: the knowledge base, the inference engine, the explanation module and the user interface. The knowledge from the expert is stored, or represented, in the system's knowledge base. The various methods by which this knowledge can be represented within an expert system are discussed in Section 6.4. The knowledge base is constructed by the knowledge engineer through close liaison with the expert and with the aid of the developer interface. In general, the inference engine manipulates the knowledge in the knowledge base to provide new information to solve the user's problem. The user interface allows the user to communicate with the system, and the explanation module enables the user to obtain justification of the system reasoning and explanation of its recommendations.

# 6.4 Knowledge Representation in Expert Systems Technology

Previous sections have summarised the key characteristics of expert systems and in this section, attention is drawn to the process of building an expert system and the

## Rule Based Expert System

technology involved. The discussion will focus on the knowledge representation that will be used in the production of a timetabling system.

Knowledge representation is concerned with how large bodies of domain knowledge can be organised into a data structure that facilitates symbolic manipulation within an expert system knowledge base. Various knowledge representation schemes have been proposed for use within expert systems such as logic, procedural representations, production systems (production rules), analogue representations, semantic networks, frames and scripts. Jackson (1986) classifies all major schemes into mathematical logic, production rules and structured objects which include semantic networks. Of these, by far the most commonly used knowledge representation scheme is production rules (Kastner and Hong, 1984; Fordyce et al 1987, Dukidis and Paul, 1990).

#### 6.4.1 Production Rules

Perhaps the simplest view of production rules is that they consist of facts, rules, heuristics and an inference strategy. The following is a simplified example that could arise in the construction of a single day timetable:

Example of a FACT: Teacher A teaching maths in first period must not be assigned to teach again in second period

Example of a RULE: IF teacher A is teaching

AND Subject is Maths AND Period is One THEN teacher A is free in period 2.

Chapter 6



FIGURE 6.1 Primary Elements of An Expert System '

Example of a HEURISTIC: IF teacher is A

THEN check if teacher is teaching in period 1

IF Yes check if teacher is free in period 2

IF Yes then swap subject/teacher.

IF not possible seek other alternatives.

The difference between rules and heuristics is not always clear. In essence, a rule is a logical statement that in a certain situation one should take a specified action which usually derives from the stated facts. The rule is therefore a plausible statement of procedure in a defined domain. A heuristic, however, is a strategy

#### **Rule Based Expert System**

## Chapter 6

which guides how the rules are applied. It is based on judgmental knowledge and often constitutes "rules of thumb" for dealing with a certain problem. A heuristic therefore involves selective use of the stated rules and can be regarded as a sort of "meta-rule".

# 6.4.2 Advantages of a Rule-Based Approach

There are a number of advantages to the design of a timetabling program as a Ruled-Based System, namely

- Many human experts (timetablers) use an implicitly rule based approach in applying their knowledge to produce a timetable with certain desired characteristics.
- Rule-Based Systems can be programmed incrementally, where additional knowledge is coded in additional rules.
- Rule-based systems can generate explanations very easily by following the path of rules that fire the reasoning process. This is of particular importance in solving timetabling problems because the expert has to be convinced that the suggested solution is plausible.
- Rule-Based Systems can have an entire expert that can be replicated at will, available around the clock, having location flexibility and sharable decision making resources thus increasing the availability of the human expert for more complex problems.

One other general advantage is the adaptability of expert systems to changes in the input, or in the constraints. The encoding of knowledge in the form of rules is especially oriented towards handling these changes.

## **Rule Based Expert System**

# Chapter 6

# 6.5 Rule-Based System Development For Timetabling

Much of the knowledge in timetable construction is of a tacit kind consisting of rules which experience has shown to be fairly reliable. It is possible to extract this knowledge in the form of rules using various knowledge elicitation techniques. In the context of expert systems, rules have been defined as a formalism for encoding knowledge (Chandrashekaran and Ramesh, 1987). A timetable resulting from the use of both tabu search and simulated annealing as described in the previous chapters will generally need some modification to accommodate a range of soft requirements. Due to the varied nature of these soft requirements, constructing a timetable using a conventional programming approach would be very difficult.

From the experience of solving timetabling problems with just the hard and medium constraints, it would be an almost impossible task to use conventional programming methods to satisfy all the possible soft requirements of every school. However, based on the experience of timetablers, who are considered to be the experts, it is possible that a method based on their approach can be converted into a computer program that will simulate the timetabling process.

In the following sub-sections, various rules are described to represent a number of generic soft requirements. In each case there is a description of the design of the heuristic for that rule.

# 6.5.1 Type of Rules

Five common soft requirements are proposed for constructing a timetable for high schools. These requirements may not all be required by any particular school, but they are available as a set of rules to provide a choice for timetablers from any

## **Rule Based Expert System**

# Chapter 6

school to create a timetable using the relevant heuristic models developed here. This is carried out by using a set of production rules which will be discussed in the following sub-sections. The soft requirements considered are as follows,

- Two subjects should not be assigned next to each other throughout the week.
- Two subjects should not be assigned on the same period throughout the week.
- For a specified pair of year bands, two subjects should not be assigned on the same period throughout the week.
- A pair of selected subjects should not be assigned on the same day throughout the week.
- A subject should not be assigned on a given set of periods on any day.

Each individual soft requirement is a 'fact' and are stored as knowledge in knowledge base files with an inference engine to resolve each rule. Each of the facts will be dealt with individually, and for each requirement a special heuristic is developed independent of another requirement. For example, the heuristic for the first requirement is that if two given subjects are assigned next to each other, then a procedure is used to move one of the subjects from its current position to a new location. This new location must not repeat the earlier assignment and will prevent the two specified subjects from being assigned next to each other over the week.

The procedure to satisfy each of the above requirementsuses a heuristic procedure which was created after discussion with timetablers. The search for a solution to each requirement follows the basic concept of a rule based approach. The structure of the heuristic is on the pattern IF <condition is true> THEN <take action or draw conclusion> and these conditions are tested until the search arrives at a situation that makes the condition true. Each procedure uses a series of swaps
between periods to achiece the required condition. At the end of each swap operation it is important to make sure that no clashes within a period appear anywhere in the week. This is checked by subroutine to remove clashes within periods without altering the position that has been set for the above requirement.

Each individual fact and the corresponding heuristic model are now discussed. The discussion will then focus on building rules to guide the search for a timetable solution with more than one soft requirement.

# 6.5.1.1 Rule\_One

Fact: Two particular subjects should not be assigned consecutively.

Rule: IF Fact\_One is required THEN Rule\_One is On IF Rule\_One is On THEN Execute Program R1\_Exe

Heuristic : This heuristic is designed to prevent two specified subjects occuring consecutively on any day. If two given subjects are found to be assigned next to each other, one of them must be removed to a location where such a situation does not occur. This is not a frequently occuring requirement, but may arise if the two subjects concerned would require students to move between remote locations.

# 6.5.1.2 Rule\_Two

Fact: Two particular subjects should not be assigned on the same period throughout the week.

## **Rule Based Expert System**

Chapter 6

Rule: IF Fact\_Two is true THEN Rule\_Two is On IF Rule\_Two is On THEN Execute Programme R2\_Exe

Heuristic: This heuristic is designed to prevent two specified subjects being assigned in the same period throughout the week. The reason for this requirement may be for example, the same teacher is required to teach two subjects. This situation may happen if a school has not enough teaching in a subject to justify a full teacher and therefore the teacher is required to teach a second subject.

This constraint is generally difficult to satisfy, and in many situations it is almost impossible. This is because the number of periods available in a week may be not much more than the total periods required for the two specified subjects. The inclusion of this soft requirement is mainly to cover subjects that require a small number of teaching periods.

### 6.5.1.3 Rule\_Three

Fact: For a specified pair of year bands, two particular subjects should not be assigned on the same period throughout the week.

Rule: IF Fact\_Three is true

THEN Rule\_Three is On IF Rule\_Three is On THEN Execute Programme R3\_Exe

# Rule Based Expert System

Heuristic : This fact means that for any given two year groups and for any chosen subjects S1 and S2, they should not be assigned in the same period. This type of condition is common in high schools because some schools require a teacher to teach two different subjects for two year groups. This situation usually appears when the school has a shortage of teaching staff and some teacher; has to be assigned to teach more than one subject.

The rule described above is a special case of Rule\_Two such that two subjects are not permitted to be assigned in the same period for a fixed pair of year bands. This rule is likely to be more commonly used than Rule\_Two because in practice teachers are usually assigned two subjects to teach to particular year bands.

## 6.5.1.4 Rule\_Four

Fact: A particular pair of subjects should not be assigned on the same day throughout the week.

Rule: IF Fact\_Four is true THEN Rule\_Four is On IF Rule\_Four is On THEN Execute R4 Exe.

Heuristic : This fact means that for a given pair of subjects S1 and S2, they should not be assigned in the same day. This type of condition may occur because of constraints on teachers availablities or to have these two subjects well distributed throughout the week. This means that whenever subject S1 appears in one day, S2 must not be assigned on that day for each year group.

### **Rule Based Expert System**

### Chapter 6

# 6.5.1.5 Rule\_Five

Fact: A particular subject should not be assigned to specified periods for all year groups throughout the week.

Rule: IF Fact\_Five is true THEN Rule\_Five is On IF Rule\_Five is On THEN Execute R5\_Exe

Heuristic : The statement says that subject S1 should not be assigned on any of the specified periods in any day throughout the week. This may be a single period or a range of periods. An example might be that a PE subject should not be assigned after the lunch break because it is not good for students to do a lot of running just after lunch. Subjects like a foreign language may be taught by a part-time teacher who cannot be available in the last two periods, hence any assignment of that subject must be earlier in the day. This type of requirement is very common in high schools and the degree of importance varies from school to school. In terms of timetabling, whenever subject S1 appears in the specified period(s) it must be removed.

# 6.5.2 Combinations of Soft Requirements

Each of the heuristic procedures will satisfy one individual requirement. A special heuristic is required if more than one requirement is needed. For example if Rule\_One and Rule\_Two are required by a school there is a problem of deciding whether Rule\_One or Rule\_Two is fired first. Another problem with this procedure is that it may cause a violation of Rule One if Rule One is fired first

### Rule Based Expert System

followed by Rule\_Two. To prevent this, a special heuristic is designed which will satisfy Rule\_Two while maintaining the validity of Rule\_One. Alternatively, if Rule\_Two is fired first, the heuristic algorithm must satisfy Rule\_One and maintain the validity of Rule\_Two. Analogous procedures are created for every possible pair of the requirements.

The importance of each requirement is measured by the rating of each individual rule in three levels, 3 being the most important and 1 as least important. For each possible combination, a heuristic program is created and is called R\*A\*\*.Exe where requirement \* (\*=1,2,...5) is less important than requirement \*\*(\*\*=1,2,...5). For example, if Rule One and Rule Two are needed and if Rule One is more important than Rule Two, then the action taken will be to execute programme R2A1. Exe. For each pair offequirements, three procedures are needed which are based on either rule having greater priority and being fired first, or both rules having equal priority. The heuristic for any pair involves the satisfaction of both conditions. The stopping criterion on the search is a limit imposed on the number of iterations. During the search, a strict condition is imposed on any swap operation that it does not violate the hard and the medium constraints and also does not override the position satisfying the rule fired earlier. This restriction automatically reduces the number of available swap options and makes the task of searching for a better solution much more difficult.

A similar procedure was investigated for satisfying three requirements simultaneously, but it was found that the procedure invariably either goes into a loop or stops without satisfying all the requirements. It was therefore decided that there was no point in trying to satisfy more than two requirements simultaneously and a different approach is required to satisfy more than two requirements.

# 6.5.3 Combinations of Three or More Soft Requirements

In practice, any combination of the 5 requirements may be involved, which may result in firing anywhere between one and five rules. For example, if a school only requires two subjects not to be assigned on the same day, then this means that Rule\_Four must be fired and no other rules. If more than one instance of the above condition is needed, then the action will be to fire the same rule repeatedly. As mentioned earlier this action may cause the swap operation to override a previous allocation. A simple principle is followed, namely that the least important requirement is fired first (LIF)and the most important requirement last (MIL).

The order of firing the programme is important because the result from the first programme will be used as the initial solution for satisfying the second requirement. The principal of the heuristic is as follows. It begins by executing R\*.Exe, a programme which will satisfy the first requirement, which is least important and uses the result to input to the second requirement, which is more important, and so on to the third and subsequent requirements. In satisfying the later requirements, it may override the allocations made when satisfying earlier requirements. To overcome this problem, backtracking is carried out where any rules not now satisfied are fired again in ascending order of priority. This continues until either all requirements are satisfied or it has reached a specified number of iterations. In the case when not all requirements can be satisfied, the lower priority requirements will be sacrificed.

There will be some cases where two or more rules have an equal importance. In this case, the order of firing the rules is not significant and each may override the others. If this occurs a backtrack is carried out by firing the other rules again.

### **Rule Based Expert System**

The success of the execution of combinations of rules is measured by a rule objective function which uses the number of rule violations as a measure. Any violation of a rule may be overcome by doing a backtracking, that is by returning to the individual rule and redoing the search to overcome each particular rule violation and the newly generated table will be re-evaluated. This is repeated until it reaches an optimum solution or completes a fixed number of iterations

# 6.6 Example and Discussions

An experiment to test the use role of production rules to satisfy the soft timetabling requirements described in this chapter was carried out using the same set of hypothetical data used in Chapter 5. This involves a timetable with 30 weekly periods, 12 subjects and 6 year groups. For demonstartion purposes, a test was carried out using timetable requirements from Market Bosworth High School. This school requires two soft requirements, namely that Design (subject 2) should not be assigned next to Games (subject 9) throughout the week. The priority of this requirement is set as 2. It also requires that PE (subject 10) should not come immediately after mid-day break in period 5. This requirement is rated as 1 which is not so highly desired.

Since the first requirement has a higher priority than the fifth requirement, the rule is to execute programme R5A1 as described in Section 6.5.2. This programme will read the initial timetable generated using tabu search and simulated annealing in Chapters 4 and 5, from which it will aim to satisfy both requirements. The procedure works by first satisfying requirement 5, followed by requirement 1 but in satisfying the latter requirement, violation of earlier requirement may occur. Violation of requirement 5 is removed by backtracking until either both requirements are satisfied or the assigned number of iterations is reached.

# **Rule Based Expert System**

# Chapter 6

			-																	
Band			Mo	nda	I <b>Y</b>				]	<b>Fue</b>	sda	V				W	edn	esd	ay	
1	2	2	8	3	4	5		7	4	4	5	9	3		5	7	5	8	2	2
2	7	7	4	8	9	3		3 (	9	$\mathbf{D}_{\mathbf{z}}$	2	8	8		8	11	3	9	5	4
3	10	9	5	8	2	2		5/	7	9	8	2	4		3	6	11	4	8	9
4	8	4	9	9	3	8		2	2	5	9	5	7		4	3	2	2	4	7
5	3	8	2	7	11	4_		9	11	3	3	4	10		9	9	8	3	7	8
6	9	3	3	2	8	7	V	4	8	6	7	3	9		7	4	9	6	9	6
Band		]	Chu	rsd	ay_	1			·	Fri	dav	,		Τ		S	atu	rda	y	
Band 1	1	ך 9	[] 10	rsd 6	ay8	f		11	9	Fri 8	day 3	, 9	4			S	atu	rda	<u>v</u>	
Band 1 2	1	ן 9_ 5	<b>hu</b> 10	rsd 6 10	ay 8 5/	/ 17 4		<u>11</u> 4	9	Fri 8 6	day 3 2	9	4	I		S	atu	rda	<b>y</b>	
Band 1 2 3	1 1 1	9 5 3	<b>hu</b> 10 7 9	rsd 6 10 4	ay 8 5/ 7	7 4 3		11 4 5	9 5 3	Fri 8 6 5	day 3 2 7	9 2 ( 4	4 9 2			S	atu 	rda	<b>y</b>	
Band 1 2 3 4	1 1 1 1	9 5 3 7	<b>hu</b> 10 7 9 3	rsd 6 10 4 8	ay 8 5/ 7 9	7 4 3 5		11 4 5 6	9 5 3 4	Fri 8 6 5 3	day 3 2 7 11	9 2 ( 4 5	4 9 2 3			S	atu —	rda	<b>v</b>	
Band 1 2 3 4 5	1 1 1 1 1	9 5 3 7 4	<b>Thu</b> 10 7 9 3 6	rsd 6 10 4 8 7/	av 8 5/ 7 9 2	7 4 3 5 2		11 4 5 6	9 5 3 4 8	Fri 8 6 5 3 9	dav 3 2 7 11 6	92(45)	4 9) 2 3 7				atu —	rda	<b>v</b>	
Band 1 2 3 4 5 6	1 1 1 1 1	9 5 3 7 4 11	Thu           10           7           9           3           6           4	rsd 6 10 4 8 7/ 1/1	av 8 5/ 7 9 2 3	743529		11 4 5 6 9 7	9 5 3 4 8 2	Fri 8 6 5 3 9	dav 3 2 7 11 6 8	924563	4 9) 2 3 7 8				atu —	rda	<b>v</b>	

Periods

TABLE (	6.2 Initial	Timetable R	showing via	olation of	rule 1	and 5.
---------	-------------	-------------	-------------	------------	--------	--------

									Per	iod	8									
Band			Mo	nda	iy				1	Tue	esda	iy .				W	edn	esd	lay	
1	2	2	8	3	4	5		7	4	4	5	9	3		5	7	5	8	2	2
2	7	7	4	8	9	3		36	7	2	2	11	8		8	9	3	9	5	4
3	5	9	7	5	2	2		8	5	9	8	2	4		3	6	11	4	3	9
4	8	4	9	9	3	8		12_	2	5	9	5	7		4	3	2	2 '	4	7
5	3	8	2_	7	11	4		19	9	3	3	4	10		9	11	9	3	7	8
6	9	3	3	2	8	9	$\overline{I}$	4	8	8	7	3	9	ľ	7	4	7	6	9	6
							Τ													
Band	ŀ	]	<b>Fhu</b>	rsd	ay	/				Fri	day	,				S	atu	rda	<b>y</b> .	
1	1	9	10	6	8	17		11	9	8	3	9	4							
2	1	5	9	10	5	A		4	5	6	2	2	8)							
3	1	3	7	4	7	8	Γ	10	3	5	10	4	2							
4	1	7	3	8	9/	5		6	4	R	11	5	3							
5	1	4	6	7	2	2	Г	8	8	9	6	6	7						<b>—</b>	

2 8

(1) 6

All violations removed.

11

9

7 2

1

6



### **Rule Based Expert System**

Table 6.2 shows an initial timetable with 2 violations of Rule\_One and 1 violation of Rule\_Two. The execution of this programme satisfies both of these requirements while maintaining the hard and medium requirements satisfied earlier.

Similar tests were carried out on different combinations of requirements and it was found that almost all the soft requirements could usually be satisfied. This shows that, in this particular case, the rule based approach is capable of satisfying the imposed soft requirements. This approach is used in the three case studies discussed in the subsequent chapter of this thesis.

### 6.7 Summary

This chapter has outlined some characteristics of rule based expert systems which provide a tool with which to satisfy soft requirements in timetabling problems. It has also demonstrated the potential of rule based systems to guide the system to fire the desired program to obtain a solution. The approach used is a hybrid between expert systems using heuristics for swapping and shifting allocations based on observation and discussion with timetablers.

The implementation of the rule base employs backtracking, composition and decomposition of rules and constraint ordering to reduce constraint violations. The efficiencies of each individual rule, paired rules and other possible combinations of rules are tested. The result as demonstrated in the example shows that it is capable of satisfying soft requirements without violating the already assigned hard and medium constraints. It shows how the use of Rule-Based Expert System techniques can modify a timetable generated using conventional programming

techniques. In the next chapter, this system will be used in solving real-life timetabling problems in several local high schools.

**Implementation and Case Studies** 

# CHAPTER 7

# Automatic Computerised Timetabling System Implementation and Case Studies

# 7.1 Introduction

This chapter draws together heuristic approaches and rule based expert systems described in the previous three chapters to produce an interactive timetabling system for generating an acceptable timetable for high schools. It is a user friendly system that is able to gather information and process it within a short time. The chapter commences with a discussion of general issues in timetabling system design followed by an introduction to the Automatic Computerised Timetabling System (ACT) that has been created. This system is based on a series of in-depth case studies requiring extensive collaboration from teachers involved in timetable construction. The approach taken in designing this system and its implementation are then discussed; and finally discussions of three case studies and the responses of the timetablers are presented.

# 7.2 The System Development

The first stage of the system development was to study the needs of timetablers from various high schools in Leicestershire. Several commercial systems have been developed in recent years, most of which are available in schools, but they seem to be not widely used in generating school timetables. Computers are very useful for printing out the timetable and other administrative tasks, but are not actually being used to generate a timetable. An indication of this is that all the

## **Implementation and Case Studies**

# Chapter 7

schools visited in this study either do not have a system at all, or do have a system but do not use it.

School	Systems Available	Timetabling Process
Woodbroke Vale High School	SIMS System	Manual
Market Bosworth High School	Not Available	Manual
Garendon High School	Nova-T	Manual
Limehurst High School	Not Available	Manual
Rawlins High School, Quorn	Nova-T	Manual
Burleigh College, Loughborough	SIMS System	Manual
Stonehill High School, Birstall	Not Available	Manual

 TABLE 7.1 List of schools visited in this study

Table 7.1 shows that all the schools visited in this study build their timetable manually. The main reasons for not using any system at all to generate a timetable for their schools may be summarised as follows.

- First, systems are often too expansive. Once purchased, the cost of maintaining the system can be expensive and much of the available software requires a high level of training.
- Secondly, the timetable generated often cannot fully satisfy the school requirements and still involves a large amount of manual work. Hence, it can take a long time to produce a timetable and sometimes too long to be of practical use. Often, the timetable generated does not work at all.
- Thirdly, some of the systems available are essentially just constructing a timetable manually. Instead of working on pieces of paper the timetable is constructed manually on a computer screen.

#### Implementation and Case Studies

This general absence of computer based timetabling suggests a possible need for a more effective automatic scheduling system. The system must be more user friendly and self explanatory, and able to be used by someone with little experience of computers. Nevertheless, the system will only provide assistance to a timetabler and not replace the timetablers themselves.

The design process began by developing a basic prototype system based on conventional programming, where all the necessary data are pre-set in advance of executing the scheduling program. The role of this prototype was to show the possibility of developing heuristic models based on tabu search and simulated annealing to generate a timetable. From initial testing, the heuristic approach used showed a valid and acceptable result (see Chapters 4 and 5). Since the system was designed to satisfy both novice and expert timetablers, the system design is based on providing as much support as possible to the users. The system took a step by step approach. The system was designed to be menu based and in almost every menu, the system will provide a help option that will allow the user to understand what is going on. The system is user-oriented and designed from the user's perspective, using common timetabling terminology.

The central theme of the design of a usable timetabling system is the flexibility of the user interface to allow the user to interact with the system. Flexibility in the user interface is very important as it reflects on the total software package and at the same time helps users to complete the task of producing a timetable more efficiently and achieve the overall goal of producing a working timetable. Nisenbaum (1994), gave three key steps in developing a system with an effective and flexible user interface. They are

- Identify the users and their characteristics.
- Identify user tasks

Involve users continuously

# 7.2.1 Users and their characteristics

A system to produce a timetable for high schools will be used by the timetabler in that school. Often the job of constructing a timetable will not involve anyone else and it is reasonable therefore only to consider the characteristics of the timetabler. The knowledge gathered from them includes their familiarity with computers and their skill levels when using computer based systems. This is likely to be influenced by the computing facilities that are available in that school. Some schools do not have any computer system that is good enough to be used to construct a timetable. In this study, a brief interview was carried out to determine the timetabler's characteristics and the approach they use for solving their timetable problems. For simplicity, the information on user's characteristics with the next step, namely to identify the current goals and the user's task.

# 7.2.2 Identify user task.

The importance of identifying user's tasks cannot be underestimated in the development of systems and user interfaces, and special attention needs to be given to collecting this information. It is crucial to identify the timetabler's goal and their subordinate tasks. This includes identifying the areas in timetable construction where the timetablers feel that a computer system would be beneficial, which areas are the most difficult and which parts of the construction processes should be interactive?. The system designer also needs to determine the level of detail that is required. The first visit to the timetabler was to understand their higher level tasks, with subsequent visits used to focus on finer details. The interactive process

#### Implementation and Case Studies

includes the display of the results generated and this also involved the users and their requirements at various stages.

# 7.2.3 Continuous User Involvement

The designer needs to demonstrate that the system will make the users' job more productive and easier to accomplish. Including users in an interactive design process can help ensure that the final product allows users to achieve their overall goals and provide appropriate facilities for users to customise the user interface. A system can then be designed to accommodate the various user characteristics that have been identified.

# 7.3 Introduction to ACT System

The ACT system has been developed specifically as an interactive system for solving high school timetabling problems. The system allows the user to create or modify the basic input data. Here the school curriculum structure, the special periods, double periods and other special requirements for each school are specified. The aim of the ACT system is to design a weekly repeating schedule that shows what, where and when teaching activities are taking place at a school. The results are output in the form of various tables which show how all classes and subjects are scheduled throughout the week.

The system uses several criteria to qualify a proposed timetable, with a variety of requirements whose relative importance may be specified by the user. The weight attached to the various requirements is a subjective factor effecting the system output globally. The solution proposed by the system based on the specified weights might not satisfy fully the expectations of the user who defined the

weights. To overcome such problems, the system is highly interactive, allowing the user to compare and manually modify the output on the screen.

# 7.4 ACT System Architectures

The system is divided into several different components, each of which contains several different sub-modules. The major components of the system are shown in Figure 7.1 and described in the following sub-sections.



FIGURE 7.1 ACT System Structure

æ,

# 7.4.1 Input Module

The input module consists of two sub-modules; the curriculum input sub-module and the expert rule input sub-module. They read in the raw data and prepare them for the processes module.

# 7.4.1.1 Curriculum Input sub-module

This sub-module reads in all the curriculum requirement's data interactively with the user. The curriculum data requirements are divided into three parts, namely the subject requirements, the double periods requirements and position of the lunch break. An illustration of the subject input screen using two letter codes is given in Figure 7.2.

· · · · · · · · · · · · ·		913.M	័ត៌ក្នុងនេះ។	eren anderstellen Alteration Alteration	alisan Alisa	ារស្តីតំ
	NOTE Subject Subject Subject Subject Subject Subject Subject Subject	S u b j e c = Tutorial 1 = Tu 2 = Ma 3 = De 4 = Sc 5 = Hi 6 = Ge 7 = En 8 = Fr	t Cod	е Nапе signed as su Subject 9 Subject 10 Subject 11 Subject 12	******* = Gm = Mu = Dr = Ea	
		****** () on od [ Select	ption tinue&Sa ifyEntry optionto	s ***** ve proceed ]		

FIGURE 7.2 Sample Screen for Subject Code Input

#### Implementation and Case Studies

Following the screen shown in Figure 7.2 the system will request the position of the lunch break. During data entry or modification of data this interval will not be displayed on the screen, but it will be displayed in the graphical output presentation.

# 7.4.1.2 Expert Rule Input sub-module

The expert rule input sub-module reads in all the soft requirement's data interactively with the user. This comprises both the soft requirements and the importance rating (weight) of each of the specified soft requirements.

# 7.4.2 Modification Module

The modification module consists of two sub-modules to permit manual modification of the requirement data and manual modification of a previously generated table.

# 7.4.2.1 Requirement Modification Sub-module

This modification sub-module provides facilities for altering the existing data. The system displays on the screen what data was entered before and may require modification. At all times, the system checks whether the data entered has violated any necessary conditions. To facilitate the entering and editing of data, subject codes are displayed as a reference, as is the number of available periods left to be allocated for that year band.

### **Implementation and Case Studies**

### Chapter 7

The system will not accept the data if the total number of periods entered is not equal to the number of available periods in a week. Any violation of this condition will be alerted automatically by the system. When all the data has been entered, the system will indicate that it is ready to generate a timetable. Modification can also be done by using any other editor but care must be taken as it does not check the total number of weekly periods automatically, and this can sometimes cause problems during subsequent processing.

# 7.4.2.2 Output Modification sub-module

The output modification sub-module provides an opportunity for the timetabler to modify a previously generated table. In other words it is a way of manipulating the existing table interactively. There are two forms of modification available. First, there is the option to swap two columns in the same or different day. Second, there is the option to shift elements from the current position to another position in the week. The procedure for this approach is described below and the subjects involved are in the form of integers as shown in Table 7.2.

The procedure for swapping two periods is a straight forward process. A period in DayOne is considered as the source period and another period in DayTwo is considered as the destination period (DayOne and DayTwo may be the same). The swap process simply swaps every element in the source period with the corresponding element in the destination period. This operation does not violate any constraints. In this module the system requires the user to provide the source period in DayOne and the destination period in DayTwo.

The second option is to move an individual element from one position to another while maintaining the condition of no repeated subject in a period. An element is identified as a source cell and a second element as the destination cell. For example, if the elements  $r_{23}=2$  and  $r_{25}=2$  then moving them next to each other will satisfy one double period requirement. An appropriate action would be to move the cell  $r_{25}$  to column 4 of the same row. In this case element  $r_{24}$  (the source cell) will be swapped with the element  $r_{25}$  (the destination cell) and this move results in a satisfaction of one double period, but a clash in period 5 on DayOne. Table 7.2 shows the result of that swap.

				Da	yOne	;			L	DayTw	0	
	1	2	3	4	5	6	1	2	3	4	5	6
1	2	7	7	4	1	5	4	8	5	2	9	2
2	4	6	2	2	5	3	2	5	3	6	1	4
3	8	3	5	1	5	2	5	6	7	8	6	3
4	1	5	8	3	3	4	3	9	2	1	4	7
5	6	2	9	8	9	7	8	3	4	7	5	1
6	7	4	6	9	6	8	9	4	6	3	7	5

TABLE 7.2: Two day timetable matrix R

After each swap, the procedure check for n possible clashes within both the source and the destination periods. In this example, a clash has ocurred  $n_{125}^{(1)}$  year band 3 with  $r_{25} = r_{35} = 5$ . The system will place all elements in that period into a tabu list and seek to replace  $r_{35}$  with other element that is not listed in tabu list and does not break any previously assigned double periods. The procedure is repeated until all clashes are removed. If it is not possible to remove all clashes, the original swap is rejected.

# 7.4.3 Processes Module

The processes module consists of two submodules: the heuristic and the rule-based expert system submodule. These are the modules which actually create the timetable to satisfy the hard, medium and soft requirements specified in the input module.

# 7.4.3.1 Heuristic Processes Sub-module

The initial timetable construction uses the method of tabu search as described in Chapter 4. As this will usually fail to satisfy all the double period requirements, further heuristics were developed to improve the assignment of double periods using simulated annealing as described in Chapter 5. The complete process is carried out in three stages as shown in Figure 7.3.

First, Otogen.Exe creates the timetable which satisfies the basic curriculum requirements. Secondly, Otodbl.Exe refines the timetable to attempt to satisfy some of the medium requirements by tabu search. Finally Otoneal.Exe improves the satisfaction of medium requirements (particularly double period requirements) by the use of simulated annealing and saves the resulting timetable as Filename.Req.

### Implementation and Case Studies



FIGURE 7.3 Processes Module Flowchart

# 7.4.3.2 Rule based Processes Sub-module

This submodule attempts to satisfy soft requirements using rule based expert systems as described in Chapter 6. This submodule takes the desired soft requirements along with their importance ratings that have been saved in Filename.Exp and Filename.Rat respectively. It then reads the existing timetable from Filename.Req and generates a final timetable guided by the rules described in Chapter 6. At the end of its execution, the generated timetable is saved in Filename.Req and overrides the existing timetable. The procedure follows the flowchart given in Figure 7.4.



**FIGURE 7.4 Processes Module Flowchart For Expert Systems** 

# 7.4.4 Output Module

The output component displays the timetables produced after the execution of the previous module. Output can be obtained either after the heuristic process (initial output) or after the expert system process (final output). These are described in the following sections. The initial output module occurs after satisfying the hard and medium constraints using the programmes shown in Figure 7.3. At this stage the timetable generated could be accepted as a working timetable or may be subject to further refinement by the expert system phase to produce the final output. The output module generates three different kinds of timetable, namely

• A class timetable. This is a 'period by day' timetable for each year band indicating the lessons at each timeslot for each day

- A subject distribution timetable. This is a 'period by day' timetable for each subject indicating the class taught at each timeslot for each day.
- A daily timetable. This is a 'class by period' timetable for each day indicating the classes taught to each year band on that day.

# 7.4.4.3 Other Options

There are two other options available within the Main Menu. A DOS option provides a temporary exit from the system to carry out any DOS commands. The INFO option provides general information on the running of the system.

# 7.5 Case Studies

This section describes the background to the case studies of timetable construction in high schools and the scope of the collaborative project that was undertaken. It commences with a brief overview of the school curriculum and the way the timetablers solved their problems. It then focuses on implementing the ACT system to produce a timetable for each school. The result from using the ACT system is compared with the manually generated timetable and followed by a discussion of the results.

The process of approaching high schools was based on contacts with the Leicestershire Local Education Authority. This initial contact proved useful as a first stage in understanding the use of computers to solve timetabling problems in Leicestershire high schools. The second stage was to obtain direct collaboration from timetablers in several high schools. Three schools were chosen for this study,

# **Implementation and Case Studies**

namely Limehurst High School, Market Bosworth High School and Stonehill High School. The timetablers from these schools were willing to collaborate and showed interest in the outcome of this study. The number of students in each school varied between 400 and 600 pupils and the students are divided into three year groups according to age (years 7, 8 and 9). All year groups follow a common core curriculum related to the National Curriculum (NC) which includes English, Mathematics, Science, Humanities (History, Geography, Religious Education), Design, Technology and Science, Music and Physical Education. As required by the NC, French and German are also offered as foreign languages.

Even though each school follows a common curriculum as set in the NC, differences do exist in the number of teaching periods allocated to each subject, the number of common periods, the number of double period requirements and in the position of day break. For example, Limehurst High School requires Science to be assigned as double periods while Market Bosworth High School does not. Each of the schools in this study has their own special requirements, and a discussion of the differences in their approach to solving their timetabling problems will be given.

# 7.5.1 Timetabling at Limehurst High School

Limehurst High School is a comprehensive school for pupils from 11 to 14 years old and caters primarily for pupils who have previously attended primary schools in Loughborough. In the 1994/95 session, about 150 pupils were admitted and divided into forms, each with an assigned form tutor who follows the form through the school for three years. In some areas of the curriculum, the forms may be regrouped depending on both individual and subject needs. Some subjects require a

#### **Implementation and Case Studies**

special room or to be assigned as double periods. Since the school buildings are divided into two blocks about one hundred yards apart and separated by a main road (see the school map in Appendix A), there are some difficulties in moving students from one block to another, and this causes complications in constructing a timetable for the school, especially when assigning subjects such as Science and Design.

All the pupils in this school follow a timetable that ensures no particular day is overloaded with subjects requiring similar skills. The process of timetabling at this school begins in April or May as by that time almost all the information, such as the size of the student intake, the number of staff available or any changes to the curriculum requirements, are known. Each year group is divided into two year bands with each pair following the same curriculum pattern with equal numbers of teaching periods for each subject. In addition, two teaching periods are spent with the form tutor in tutorial work and one period is allocated for assembly. Other subjects, such as health education, information technology, economic awareness and environmental science, are also required by different year groups. Once all the information has been gathered, the timetabler begins the process of constructing the timetable by first assigning the subject with least freedom of movement. Once the most difficult subject has been assigned, the timetabler will then assign the subject with next least freedom of movement. This process follows until all the subjects have been assigned. Producing a complete timetable manually is a long process which very often must be done repeatedly, and there is no guarantee that at the end of the process, a working timetable is obtained. At Limehurst, the timetabler estimates that it takes about 100 working hours to produce a workable timetable.

The overall curriculum requirement for 1994/95 at Limehurst will follow the pattern listed in Table 7.5 with subject codes listed in Table 7.6. The location of the two school buildings requires some time: for the students to move from one building to the other and this factor must be considered. Two subjects, Science (Sc) and Design (De), require double periods and both must be taught in a special room.

YrGrp	Tu*	En	Hu	Sc	De	Ma	MI	Pe	Mu
7L	3	3	4	3	3	3	3	2	1
<b>7</b> H	3	3	4	3	3	3	3	2	1
8L	3	3	4	3	3	3	3	2	1
8H	3	3	4	3	3	3	3	2	1
9L	3	3	4	3	3	3	3	2	1
9H	3	3	4	3	3	-3	3	2	1

Subject Code	Subject
Tu	Tutorial
En	English
Hu	Humanities
Sc	Science
De	Design
Ma	Mathematics
Ml	Modern Language
Pe	Physical Education
Mu	Music

TABLE 7.6 List of Subject Codes

Since each of them has three periods, only one double period can be assigned to each year band and the other is a single period. The tutorial periods (Tu) consist of

two periods for tutorial and one period for interview or assembly, which are fixed in the first period on Tuesday and the first two periods on Thursday. All year bands must be assigned to have tutorials and assembly in the same periods each week. Besides the basic curriculum requirements, this school also has other soft requirements as listed in Table 7.7. It requires Mathematics and Design not to be assigned in the same period for certain year bands and Physical Education should not be placed in the first period. The rating for Rule 3 is set as 1, a requirement that is low in priority and Rule 5 is set at 2, a requirement that is low in priority and Rule 5 is set at 2, a requirement with high priority.

Rule	Types of soft requirements	Responses
R1	Two subjects should not be assigned next to each other throughout the week.	No
R2	Two subjects should not be assigned on the same period throughout the week.	No
R3	For particular year bands two subjects should not be assigned on the same period throughout the week.	Yes
R4	A pair of selected subjects should not be assigned on the same day throughout the week.	No
R5	A subject should not be assigned on a give set of periods.	Yes

TABLE 7.7 Soft Requirements At Limehurst High School

A demonstration of the ACT System was carried out at Limehurst High School. The system was used to produce a timetable for this school based on the curriculum requirements and the soft requirements listed in Tables 7.5 and 7.7. The timetable produced by the ACT System is given in Figures 7.5, 7.6 and 7.7. The result shows that some of the subjects appear more than once in a day. This is due to the need to satisfy double period requirements and also due to the reduction in the freedom of movement for the subjects because three periods have been reserved for common tutorial periods. Only 23 periods are available to be distributed between the 9 subjects. This has been recognised as one of the problems in producing a manually generated timetable, which resulted in the violation of non repeated subjects in any day and, in some cases, the hard constraint that no subject must be assigned more than once in the same period.

F1 Inf	ACT oF2	S Y S dit	STE F3	M∶I View	lodifi F4	cation Save	Module F5	Çait	
			Curros	nt U-	eti				
			ំណែ	sas v. Yil∈ni.	L G TI				
			- <b>1</b> 9	12	Ω.	191 <del>9</del> .			
	Oberes De	лан. Пре	i fin:	1.17	E.				
	्यात्रा स्टब्स् संस्थित स्टब्स्	1		. n.	1971	inir Ethe			
	ារសារ សារ ស្រុកសំណ៍សំ	1	2048 704-	1.11	ារ ភូមិស្រុ	1981			
				i a ll Li line	11.	ារដោយ ស្រុក			
		110) ab-		- 1 <b>5</b> 10	×¶⊷ .T⊾	<u></u>			: 4
	The she	<u> i i i i</u>		2.0	Tu				
	HEILU AND	1)p	1)A	ង្ហា	'¥:	1 <b>11</b> 1			·
· · · · · · · · · · · ·	· ·			ിന്ന ജീബ					
		Û.	77	វិហ៍	ĩΥ	) T			
	<b>Überdi</b> 201	Ĩ	С.	Ê.	Ð.	Πh			
		2016	ille:	ាត	S.,	3.			
	Broth SD		136	СЬ.	ab-	CT.			ŗ
	End Partie	111	Űh.	1	Ŭ.	Пh.			
	Perce an	in.	Ch-	ilin_	100	Ch-			
	100000 2022 100			1011 101-	<b>9</b> 52				
	(18 × 30)	FUE	i g pi i	141-	I TERE	्मा	1913/2 1 Jacked words 7 19		
		26766	د اړل ا	. 110.	:				

FIGURE 7.5 Timetable For Limehurst High School: Monday and Tuesday

### **Implementation and Case Studies**

Chapter 7

		- Cum	ont II	1611 -		
		ourr		164		
			in the	. 17	1.	
2 <sup>9</sup> , 1 , 5 í	÷.	i.	ta	16	<b>1</b> 0	
: teste filt	á a		21	1	-14	
ាត់ព័ត៌	<b>H</b>		ja,	() :	1	
di d	9	.: F	3	4		
ीस्तामं े	Ϋe a	5				
17.116				¥1	ù,	
			<b></b>			 
		<b>D</b> e				 
		- Heri Des	i Transa	· · · ·		 
		: - (445) 	e Porte Silikada Silikada	· · · ·		 
:1577	<u>終</u>	i deve Biri W		- 		 
្សត្ថរក្ស (ត្រូវក្ស) (ត្រូវក្ស)	度 第二章 第二章		e torr Sitter RC The The			 
: 5505 5506 1506	度 第一篇 第一篇		i Tan Silkus Ro Ta Ta	a A A A A A A A A A A A A A A A A A A A		 
(666) (666) (666) (666) (666) (666) (666) (76) (7	· 一般 自品 制 推 一般 一般 一般 一般 一般 一般 一般 一般 一般 一般			<b>新新新新生</b>		 
្សេះអាច ១៩ភាច ប្រភាព ១៩ភាព ខ ១៩ភាព ខ	·	(1)。 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	e Porra Sategos State St			 

# FIGURE 7.6 Timetable for Limehurst High School: Thursday and Friday

Once all the necessary data had been entered, the ACT system took less than 2 minutes to provide a feasible timetable. The generated timetable managed to satisfy the double period requirements, but the position where this double period is allocated was sometimes not very suitable because of the need to move students from one block to the other. Several of the unsatisfactory allocations were identified and modified using the interactive module. This option in the system is essential to the timetabler because there will always be something that must be altered manually during the course of constructing a timetable, and sometimes it will be necessary to reallocate a subject to another period during the course of the school term. In a totally manually produced timetable, changes cannot often be considered once the timetable is set because reallocation will usually involve shifting other subjects and the consequent changes become difficult to

accommodate. The implementation of the ACT System to create a timetable for Limehurst High School was considered successful by the timetabler concerned.

# 7.5.2 Timetabling at Market Bosworth High School

Market Bosworth High School is a medium sized high school which caters for pupils who have previously attended primary schools in the Market Bosworth area. It has about 600 pupils aged between 11 and 14 years.

Several meetings took place with the timetabler at Market Bosworth and full cooperation was given in providing the necessary information to produce a computerised timetable. The timetabling requirements at this school do not vary much from year to year and for the year 1994/95 the curriculum requirements are given in Table 7.8.

YrGrp <sup>*</sup>	Tu	De	Sc	RE	En	Ma	Fr	HG	MU	Gm	СМ
<b>7M</b>	1	4	3	4	1	4	3	4	4	1	1
7B	1	4	3	4	1	4	3	4	4	1	1
8M	1	4	4	4	1	4	3	3	4	1	1
**************************************	1	4	4	4	1	4	3	3	4	1	1
9M	1	3	4	0	3	3	4	4	5	1	2
9B	1	3	4	0	3	3	4	4	5	1	2

 TABLE 7.8 Curriculum Requirements For Market Bosworth High School

The timetabler begins the construction of a timetable with an analysis of the structure of the curriculum followed by a discussion with the head teacher

regarding the number of staff required to satisfy the curriculum requirement. There are 7 major subjects namely Mathematics (Ma), Science (Sc), Design (De), History/Geography(HG), English(En) and French(Fr) and a number of minor subjects, namely Religious Education (RE), Physical Education (PE), Music (Mu), Drama (Dr), Games (Gm) and German (Gr). This school has 6 equal daily periods with one period set as a tutorial period, assigned as common period for all year groups in the first period on Thursday. Among these subjects, De should be assigned as double period but since year 9 requires only 3 periods for Design, this year group will have only one double period. It has been the practice at this school that the staff are allocated to the subjects in advance and subjects like Design are timetabled first. The students in this school are divided into 8 forms within each year group which, for the purpose of timetabling, are divided into two year bands with 4 forms in each band.

As at Limehurst, the assignment begins by first assigning the subject that has the least freedom of movement. Once that particular subject has been allocated, the rest of the timetable is completed by assigning the subjects one by one until all of them are in place as required by the curriculum. The difficulty in sorting out the timetable lies in the allocation of minor subjects such as PE, MU, RE, Gr and Dr. In order to simplify the construction of the timetable, these minor subjects are combined with other subjects and assigned as one 'subject' with a modified curriculum. An example of this situation is the subjects English and Religious Education which are assigned as one group which is coded as RE. This means that, for example, one class will have Religious Education while other classes are having English Likewise Dr, Mu and PE are coded as MU and the remaining subjects are coded as CM. The manually generated timetable given in Appendix B demonstrates the allocation of these minor subjects.

# **Implementation and Case Studies**

### Chapter 7

Besides the basic curriculum requirements, this school also has other soft requirements as listed in Table 7.9.

Rule	Subject(s) involved	Priority		
R1	Design and Games not to be assigned next to each other.	2		
R5	Pe should not be assigned on period 5	1		

TABLE 7.9Soft Requirements and its priority status forMarket Bosworth High School

The requirements given in Tables 7.8 and 7.9 were used to generate a feasible timetable using the ACT System, which took less than 3 minutes. Several timetables can be generated if necessary and manual adjustments are usually required to solve some of the minor subject requirements described earlier. After several runs, the system managed to satisfy all the double periods required by the curriculum. The generated timetable is given in Figures 7.8, 7.9 and 7.10 in the form of a graphical display of the daily subject distribution.

# **Implementation and Case Studies**

# Chapter 7

F1 Info	≕ACT F2 :	SY: dit	STE F3	liew iew	lodif F	icati 4 Ca	on No ve	dule: F5	ait		
	····		Curr	ent V	iew —	•			-	·	· · · ·
			6	ifike							
		15	1	1	. <b>1</b> 4 -		127				
	电面面 猪	103	.ìr.	He	- h	111	់ជា				
	ीत्तातः य	្រុំត្	ें।	ίn.	8.	: (fé	i h				
	ing the second	Цų.	15h	ie.	ិវារ	- R	lit				
	មិនចេត មិនិ	( <b>)</b> :	Ē.	Ϋ́.	СŤ.	1	1.1				
	<b>Ümni</b> Sh	Ū.	- fle	×,	lle	67	E.F.				
	Santa (91)	(A)	×.		ir.	ሚከ።					
				- Isra							
		ŵr.		4.05.05		.77					
	in the second states	1911 11 - 1	5.2	100	17.5	100	। सरू त्रीक				
	<u>6500 46</u>		li e a	10	:	ខ្មែ() ( សិង្ហារ					
	lifetitos 7463	6046 	2.6				i lite				
		35	- 100 C	1	unte: ave		110				
	Hame BH	112		121		10	-46				
	धुत्रातः अपि विद्याति अपि		् <u>र</u> ुः भाष	ite.	S. St	36 36	ίų. Ch				
ngan na ang mang mang mang mang mang man	· · · · · · · · · · · · · · · · · · ·							· · · · · · · · ·		·····	

Select Gatana . . . f

FIGURE 7.8 TimetableFor Market Boswroth High School: Monday and Tuesday.

F1 Info	ACT F2 Fd	SYSTI lit F3	M:M View	odifica F4	ation Mo Save	odule= F5	alt
		UUP1	CHC VI	5V 1.			
		ារ។ ជីវិត ជាត	្រាមអនុរហរ ហិរិទ	់ មិ¥េរំ	ารี บาร		
	in it.		1.	ang a	an ian De ian		
		্রাদ এজ আঁচ আঁচ	1011 701-		ណ្ដូន ជាដូន ឆ្លោះ ជាដែ		
	85100 AB	ាទ ហុទ	10	10 D	1917 - 1918 De - 1916		
	ម្មរដ្ឋ រករាទរួម			590 - 11 S			
		ាម ស្ត្	HIP M	្ន			
	literation and a state of the s	the time	6.5		ar afte		
	6, जन्म भाष	ាគ	l (fi	13.	81 (H)		
							·····
, بر میچه است. بر این و می است. بر این این و می و این و می این و می و این و می این و می و ا			i iktao	- der als Madelaistat		·	
		13	madee				
		136 192	110	£2s i			
	ilanni 28	in lin	(He)	fil i	le ilt.		
	Benni 78	ີ້ພາ ເມື່ອ	ž.	ên.	ক টাই		
	Paril II	តវីរ សហ	1311		A. 184		
	ອີ້າກາໃ ບໍ່ອີ	in fr	100	in r	វិច ដែ		
•	in in	រិត មិន		15			
	900 G 00			5 - 5			
		· · · · · · · · · · · · · · · · · · ·					
	\$	elect Out	166	1			



Wednesday And Thursday

# **Implementation and Case Studies**

	F1 Info	<b>F2</b> T	S Y : dit	STE F3	M :   Siew	lodifi F4	icati I Ca	on Mo ve	dule= F5 🔅	uit	
				Corr	ent V	ieu -	· · ·			· · · · · ·	
				41-3	ាំពានហើ	: <b>1</b> 1					
				ΡŢ	12	Ξĸ.	1	Û.			
		ikan. A:	Ń.	13.	31	ffe	Ĥ.	ŵ.			
		ปีสาวกะ นี้ไม	P.	in.	11.	ា	5	1			
		Brain 200	3	ίų.	ИŤ:	a.		6.4			
		र्गमन्द्रः स्पि	Đ.		- îr	i.	If	ñ			
An A A A A A A A A A A A A A A A A A A		ীদ্যার প্র	1	ŝ.	1	N.	Ñ.	वाह			
Upp: Waa- Inning Innin Inninning Innin Innin Innin Inninnin Innin Innin Inninnin Innin		Starte Al-	L,	-tri-	h	3	Э:	Ê.			
1.011fr, 560 - 512 - 542 - 546 - 540 - 521 - 521 - 5112		Bann An Bann An Bann Gill Dann Gill Dann An	a carrow confe	3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-3-		नेजिन्द्रां हो हो	et 등은 음 등 등 등	<u>렮콽꽇묷</u> 뢒팶뫶			
						· • • • • •					



The automatic scheduling sometimes generates a result which is not very suitable, such as assigning two Mu in year band 7B, one in the first period and the other in the last period on Monday. Also it assigned two History/Geography in band 9A on Monday. The manual modification of subject allocations was demonstrated to the timetabler to enable him to try and resolve the unsuitable allocations. If he is unable to do this manually, a new timetable can quickly be generated. This is done by returning to the main menu and repeating the automatic scheduling option without having to quit the system. For this case study, several timetables were generated, one of which is displayed above. Comparison between the manually generated timetable and the ACT system timetable shows very little difference. The timetabler at Market Bosworth confirmed that he would be happy to accept at least one of the generated timetables in place of the one he had produced.

# 7.5.3 Timetabling at Stonehill High School

Stonehill High School and Community Colllege is an 11- 14 high school which provides education for over 520 children in a wide range of subjects and activities. Stonehill operates a common core curriculum for all pupils with extra subjects such as German, French and Information Technology taught to the relevant year groups. The widest possible range of subjects is offered. As with other high schools, no selection between subjects is possible and students from each year group must follow the same curriculum.

The grouping of pupils into classes has influenced the way the timetable is constructed. Each year group is divided into two year bands with a varying number of classes in each year band, and no two year bands have the same subject in any period. From the interview conducted with the timetabler, this school has enough teachers and rooms to satisfy the curriculum.

Stonehill High School follows a five day week cycle with eight periods per day. The time for each period is 40 minutes for the periods before the lunch break and 35 minutes for the periods after the lunch break. The balance between subject areas is quite complicated and a decision as to the curriculum to be adopted can only be reached after full discussion by all the staff. Before a timetable is drawn up, at least 10 meetings of staff are held to decide the suitability of each subject allocation, and this could easily take about a week. From these meetings, a timetable is constructed after all the necessary compromises are made. The Stonehill pattern was based on equal rations of time for each of the 8 subjects,
which cannot always be satisfied. Like other timetablers, the assignment of subjects begins with the subject with the least degree of freedom.

Subjects like Music(Mu), Religious Education(RE) and French (Fr) are grouped together. This is because, within each band, the different classes will have their lesson at a different time as there is a limited number of teachers to teach these subjects. There are different requirements between year groups for the Modern languages subjects (French and German). If German and French are taught, then all year groups will have 3 periods each, but if no German is taught then for years 8 and 9 there are 5 periods allocated for modern language and 4 periods for year 7. Music requires only 1 period for years 8 and 9 and 2 periods for year 7. One possible curriculum requirement for the year 1994/95 is given in Table 7.10.

YrGrp	Ma	En	Sc	De	Hi	Ge	Gf	Pe	Mx
<b>7</b> S	5	5	6	6	3	3	4	4	4
<b>7</b> H	5	5	6	6	3	3	4	4	4
8S	5	5	6	6	3	3	2	4	6
<b>8H</b>	5	5	6	6	3	3	2	4	6
<b>9</b> S	5	5	6	6	3	3	2	4	6
9H	5	5	6	6	3	3	2	4	6

**TABLE 7.10 Curriculum Requirements for Stonehill High School** 

Since there are eight daily periods, it allows for much more freedom of movement and many of the subjects are able to be allocated next to each other as double periods. Due to the large number of daily periods, the soft requirements are able to be accommodated manually and thus none of the soft requirements proposed in this system is needed. Two subjects that are required to be assigned as double periods

#### Implementation and Case Studies

are Science (Sc) and Design (De). Other subjects may also be assigned as double periods but satisfying them is not essential. One of the features in the manually generated timetable is that it seems that almost all of the subjects are assigned as double periods. The manually generated timetable for this school is given in Appendix C and the ACT timetable is shown in Tables 7.11, 7.12 and 7.13.

## Implementation and Case Studies

## Chapter 7

F1 Info F2 I	S Y dit	STE F3	H:∣ ∷iew	Hodif F	icati 1 Sa	on Mo ve	dule= F5	luit	
		Curr	ent V	ieu					· · · · · · · · · · · · · · · · · · ·
		1.15	Til A						
	115	15		н У <sub>1</sub> .		147		1.4	
出行的 劝	6:	St.	4	_∰.	! <b>!</b> ₿.	3	17.	( <b>h</b> )	
Brith M.	- <b>1</b>	· 31	ΞÌ,	a),	· P.	ti-	- Er	ê.	
Barrol (M			Η.	ເຊິ່ງ.	ħ.	ây.		<u>131</u>	
iterrati Hite	ិត	*	: 2	i sr	ាក	ΠĘ.	15	រើវត្	
	۵),	iP:	6.	5	ំពីត	8.	: <b>.</b> .	. 6	
सुरेशन्तरी संसि	(9);	57	an		÷.,	10	17	ñ,	
	.74	199	i litra asrijaji		ar.	÷.	1.00		
	141			140			1-7	1.44	
	- 112 	135	H:	1175. Gla		12			
	5 <b>1</b> +		1	1054		R.	,	127) (127)	
		10) 13-	1.0	118. 184	1910	an Or	1100 Area	S.F.	
Band BB	2943) 1945	100 E	- 11	- 1117) - 1117)	10.0	1	<b>H</b>	(B))	
		10	<u>п</u> .	int: Te	10	104	- 10 - 10 - 10 - 10 - 10 - 10 - 10 - 10		
estite as	· · · · · · · · · · · · · · · · · · ·	1915.	- 31	1 <b>-</b>				- 1010 - 1010	فالفو التربي وخوان المتكرات فالتك وحدما المكتمو والمحد والم
	) Kisi ne	1 ft 1	i		t				



F1 Info F2 F	SYS Edit	TEN F3	l:M iew	odific F4	ation Save	i <b>Ho</b> d	ule= F5	Quit	
		Curren	E Uio	:0					
		Math	(ATTEN	í,					
	13) 131		.*	Î₽4		P.	1.14	1110	
ंडाल्य यो	( <b>P</b> =	1 Ki	Щe —	រៀង	1814		131	i lin	
Brand 738	វី ត្រ	йb.	н.	·n.	alla,	វិត	ហ៊ុះ	រៀច	
Band 80	80	9	÷ <del>R</del> i	131	( and a second s		8	( <del>6</del> 7)	
Bend BB	i de la companya de la	1 an	3	S.4.	i) n	i) o	19:	្រែន	
SR forest	: 6 :	9 <u>0</u> 0	120	), in the second	(e):	P.	137	ាំខ្ម	
ि सिंहानुसं आग	10 <b>1</b> 2	H.	÷.	l'h	( <b>1</b> )-	10	1.1	18	
		(Dec.	ika- sélar						
	1,41	3/2	138	124	115	145	1	<b>B</b> S	
	115	i k	2 <b>8</b> -	10 <b>7</b> 0	nic –	13		110	
			4		.e	16	E.	115	
	2	5	: <b>5</b> )	815		1	ι III)		
	10 10		<u>' 1</u> 5	1	20	н. 5	187	n na Al	
	5	1 1	(1). /a~	ाम् । जिन्	1	147. 	100	2	
SPUTE AB	11:5	n hereinen in er.	n.			· .¥ 🖂			ana alian dalam sekarikan kelarakan sebuah kel
and a star of the star and the star of the	Carloret	<u></u>					<u> </u>	مع مراجع الم ومناطقة الم	

FIGURE 7.12 Timetable For Stonehill High School: Wednesday and Thursday

#### **Implementation and Case Studies**

#### Chapter 7

							_		
0.000	сонносникова С	H A N	GE	J I E	<u>П</u> (село	<u>(4(4), (4(4)</u>	0 kJa		
· · · · · · · · · · · · · · · · · · ·		- Ca	rrent U	iev	· · ·				
		1	10161-0						
	iî.	i D	с т <b>і</b>	· .	12.	11	Ţ	<b>T</b> .	
	mente Ro (k	- 4		1.)-	í Ye	(a)-	ů.	ju-	
		ia (n i- în-		144. 176	146-	аца. 1911 -	1.1		
			1. 1.		110.	114	24	<u></u>	
	ច្រុំជាល អាម ខ្	й (н	116	112	ar.	a de c	36	1.17	
	Rome Hills - Ig	F. 4 <u>1</u> 6		136		× .	91-	Line.	
	क्षेत्रत्ताः असः 🛛 🗄	ti Ri	) मे		554	110	н.	Ĥ.	
	ि विणि भंगतां ही	ក្ 🖓	a 🕂	H	137.	i));	2	HL.	
			Sec. Inc.						
			Nation when	·.					
	ii	6 6		Treas	1014	ill'es		1 <b></b>	
	iner an airth	• 187		10.4		мт.		) M 1 '	
1									
	ience wie								
	Hannit 18(A)								į
	Band BBB								:
:	ीर जितनी								
	कित्तती और 🔜								: *
E L L 1 Cumpont		3.22 1	Nove III	011	ŕ i	. 1 T	lo tum	to monu	
	016W L	17. 1	MGX6 V.	1.01		o I I	ecuri		

FIGURE 7.13 Daily Timetable For Stonehill High School: Friday and Saturday

The timetable generated using the ACT system does produce a timetable with better spread than that produced manually. One of the results from the discussion with the timetabler at this school is that having almost all subjects assigned as double periods is not a very good idea. This is because the students have to be taught almost all the subjects for 80 minutes if it is before lunch break or 70 minutes if the subject takes place after lunch break, and this was felt to be too long for many subjects. The system is capable of producing a timetable as an alternative to the present timetable and repetitions of subjects in the same day are kept to a minimum.

## 7.6 Summary

The first part of this chapter described the implementation aspects of the Automatic Computerised Timetabling System. This system proved capable of producing a

#### **Implementation and Case Studies**

sufficiently good timetable as shown in the three case studies that were carried out. Because of the similarity in the curriculum requirements at high school level, this system seems to have the potential to be used by all the high schools in Leicestershire. One of the advantages of this system is that it can generate a new timetable within a short time and the system has the capacity to allow the user to modify the timetable generated. Since it is based on random search, every execution will produce a different timetable and so the timetabler can choose one that best suits the school.

A solution to the timetabling problem can usually be found, particularly if each soft requirement involves only individual year bands and individual subjects taught in specific periods. The system cannot promise to provide a solution to every problem but it can be used to assist the timetabler. A copy of the system was given to each school in return for the support provided in completing this project.

## **CHAPTER 8**

## SUMMARY AND CONCLUSIONS

## 8.1 Introduction

This chapter briefly summarises the material presented in previous chapters and then discusses in further detail some of the results and findings, in particular, the methods used to develop an expert system for solving timetabling problems. From these observations some tentative conclusions are drawn concerning the appropriateness of expert systems in some aspects of timetabling problems. Finally, some thoughts on possible directions in which future research in this area might be pursued are offered.

## 8.2 Overview of The Research

The foundation of the research arises from a comprehensive review of the literature on computerised timetabling and the use of rule based expert systems in timetabling.

The review in chapter 2 showed that, for a number of years, various attempts have been made to construct school timetables using computers. At one level, the use of computers to assist the clerical aspects of timetabling leads to an obvious benefit. In addition, several mathematical modelling approaches have been proposed, based on integer programming, graph theory and also heuristic approaches. However, research has shown that there are various theoretical and practical problems with mathematical modelling approaches, and that even the more adaptable heuristic

#### Summary and Conclusions

approaches have limitations in solving timetabling problems. As a result of these limitations, there is no universally agreed approach to date for achieving an optimal timetabling solution. Due to the complex nature of timetabling problems, more often timetablers resort to finding an acceptable solution rather than an optimal solution, although in many cases it is difficult to define what would constitute an optimal solution.

Furthermore, from discussions with many timetablers responsible for preparing school timetables, there is some doubt that the methods described above could be used to 'solve' their timetabling problems. One such doubt arises from the fact that most available systems concentrate on solving a fixed set of hard requirements and play down the need for other softer requirements. Yet in many timetabling situations, these soft requirements are prevalent and these are areas which often cause a lot of headache in finding a good quality timetable. Thus, it is important to realise that solving a timetabling problem involves a series of interrelated decisions, and that few timetabling situations are exactly the same. A soft requirement for one school may become a hard requirement to another. The reviews in chapter 2 showed that the usual way of solving this issue is by decomposition of the problem to solve all the hard problems first, followed by the medium and soft requirements at a later stage. In many real life timetabling situations, which requirements fall into the hard category and which will fall into the soft category is not clear cut. It is possible to conclude, therefore, that radically different approaches to timetabling are worthy of investigation.

Expert systems are a relatively new area of computing that have evolved from research into artificial intelligence. In essence an expert system (ES) is a computer program that aims to capture the expertise and simulate the performance of one or more human experts in a problem domain. This expertise consists of knowledge and judgement, for example loosely associated facts and rules, often learnt from

#### Summary and Conclusions

experience rather than theory. This expertise is frequently formulated as simple heuristics, or rule of thumb, and is predominantly qualitative in nature rather than quantitative. From this review, it was suggested that a rule based system could be used to prepare a timetable in a high school. The main concern in using this approach is to satisfy those requirements that are considered to be soft requirements. As mentioned earlier, the general nature of soft requirements is not clear, so the proposed approach allows for a choice of requirements. Each of the requirements has a certain degree of importance and can be classified on a simple ordinal scale. It is concluded that the process of rule based expert system matches closely to the type of action taken by a timetabler in preparing a timetable, and that an expert system may be appropriate for improving the quality of a timetable. Based on these premises, this thesis has sought to explore the possible roles and benefits that expert systems can provide in the area of timetabling.

In particular, it was suggested that expert systems have several characteristics that may lead to improved quality in the timetable output. Firstly, expert systems emphasise qualitative rather than quantitative knowledge. Secondly, as timetabling is an area where relevant expertise or knowledge is generally dispersed amongst several timetablers, then expert systems provide a means of consolidating these multiple sources of expertise within a single knowledge base.

Several important issues in the proposed solution are also discussed, including the use of an objective function comprising several components in the search for an optimal solution. Since each of the components involved in the objective functions are not of the same stature, their relative influences are determined by the weights. The higher the degree of importance of the requirement, the greater will be its weight. The problem is again solved in sequence, with the hard constraints satisfied first, followed by the medium and finally the soft constraints.

### 8.3 Review and Evaluation of Case Studies

Contact was initially established with a number of case study schools to explore the possibility of a collaborative project in computerised timetabling for high schools. Timetablers, who are mostly deputy head teachers or heads of department, were convinced of the possible benefit that a computer timetabling system could provide and were willing to collaborate in the project to develop a computerised timetabling system for their school. The timetablers saw two major benefits from this collaboration. Firstly, the potential use of computer technology in assisting the work of timetable construction. Secondly, the timetablers were keen to review their method of approach and possibly learn new approaches for solving timetabling problems.

Case studies were conducted with timetablers who were willing to collaborate fully in the project. Various differences were found in the way the timetable was constructed and the type of requirements involved. These differences were taken into consideration and included into the design of the ACT system. One of the common features in the way the timetable was constructed is to make modification based on the previous timetable. If a timetable is reconstructed from scratch, then any feasible solution will be accepted and modification of this timetable often takes place during the first few weeks of the term. An automatic computerised timetabling system would be highly beneficial in such situations.

## 8.4 Discussion and Evaluation of Heuristic Methods Used

An investigation of heuristic approaches for solving timetabling problems has been one of the main objectives of this research. The two techniques for solving combinatorial optimisation problems which currently received much attention are

#### Summary and Conclusions

simulated annealing and tabu search. Both of these methods were studied and heuristic models based on these approaches were used to solve a timetabling problem for high schools. The main issue in using these methods is to study the applicability of both methods in solving an NP-hard problem such as timetabling.

The important issue here is to generate a timetable good enough for a timetabler to use. This means that either the generated timetable could be implemented fully or with little modification. The main constraints are that no subject is allowed to be assigned twice in the same period, common periods must be fixed to a particular period where every year group must follow the same lesson and the assignment of day breaks must be fixed in advance. The next constraints that must be satisfied are the requirements for double periods, where a specified subject is assigned in neighbouring periods.

Tabu search and simulated annealing were used in an attempt to satisfy these requirements. Tabu search proved to be an excellent technique in searching for the optimum solution but in some cases it could not fully satisfy every double period requirement. The failure to satisfy every double period requirement is principally because so many periods have been blocked once a double period is assigned. The use of simulated annealing based on the results from tabu search was usually able to improve the assignment of double periods.

The overall objective of the search was to establish an acceptable timetable output to satisfy the hard and medium constraints. In satisfying the hard constraints the objective function can be set as follows,

 No subject is assigned more than once in the same period and all the curriculum requirements are satisfied.

#### **Summary and Conclusions**

The first of stage is to establish a heuristic method to satisfy the above objective. If this objective is achieved, then the timetable may be accepted as a feasible timetable. Generally this is the key constraint, even though in some schools occurrence of the same subject in one period is allowed. Besides satisfying this objective there are other operational objectives that could be included, such as

- Certain specified subjects are to be allocated as double periods.
- Subjects are not to be assigned more than once on the same day for any class/year band, other than as specified double periods.

This last objective is to allow for a well distributed allocation of subjects throughout the week. At the end of the search operation, this objective can usually be achieved provided the school does not require many double period allocations. If double periods are required then it will tend to reduce the degree of freedom for allocating other subjects and will often cause a subject to be assigned more than once in the same day (apart from the obvious double period subjects). Nevertheless, from the tests carried out, the results achieved are usually acceptable to the timetabler.

Additional requirements vary from school to school, these are the soft requirements. Common soft requirements are formulated and the different degree of importance of each requirement is decided by the timetabler from each school. The method used for solving these requirements is by a rule based expert system approach. An inference engine is developed by modelling the soft requirements as a set of rules with associated heuristics. Each of the rules is weighted according to the degree of importance of the rule, and use of the heuristics is based on the principle of least important first (LIF). The reason for adopting this approach is to

#### Summary and Conclusions

avoid an important requirement being overridden by a less important requirements. Firing the most important last (MIL) will usually assure the most important requirement is satisfied.

After each rule is fired, the requirements are evaluated by using a specially designed objective function to determine the satisfaction of all requirements. Any rule violation is dealt with by refiring the heuristic for the violated rule and which will generally reduce the amount of rule violation. The results from various case studies shows that the rule based approach managed to satisfy almost all soft requirements

## 8.5 Discussion and Evaluation of the ACT System

ACT is a timetable generation system based on the timetabling structure of high schools in Leicestershire. Nevertheless, it can still be used for solving non-complicated timetables for secondary schools generally. The design of the system is menu based and interactive, with graphical user interface. This interactive feature is important for several reasons:

- A menu facility makes the user interface very straightforward. It allows users to invoke any of the options available and return to the main menu when the operation is over.
- Many of the end users of the system will be timetablers who may not have much prior exposure to computers. It is hoped that this interface will help them feel at ease when working with the system.
- Many of the on-going activities can be relayed to the user. This includes the process of subject allocation which can be seen displayed on the screen.
- Often the automatically generated timetable will not satisfy every school requirement. Modification options to the generated timetable are essential.

This system allows such modification to take place and was found to be of great help to end users.

These facilities that are available in ACT make the system more user friendly and keep the user in touch with the system. They also allow the user to make amendments to the data or the timetable as they go along.

## 8.5 Future Work

The research presented in this thesis has been both exploratory and experimental in its nature. It has raised several important questions and these indicate possible directions in which future work in this area could be pursued.

The difficulty in generalising the timetabling problem for all high schools has meant that the case studies could only be pursued in depth for schools with a common curriculum structure. The majority of the previous methods for solving timetabling problems have focused on solving the hard constraints and most of the soft requirements are dealt with manually. Soft requirements in timetabling are so difficult to generalise because of the variations encountered. Thus one possible direction for future work would be to create an editor where users can add new requirements to the existing set of rules. This calls for the creation of an expert system shell based on the methodology described in this thesis for solving high school timetabling problems.

The research has highlighted the potential of rule-based expert systems to guide the search for the best heuristic to satisfy the specified requirements. A future expert system would consist of an inference engine which is capable of generating and testing the timetable by being able to interpret and follow recommended rules during assignment. The expert system would involve the allocation of subjects

#### Summary and Conclusions

based on imitation of the whole process of subject allocation by timetablers. This knowledge will then be kept as data from which the rules are built. Therefore an editor to enter and modify rules should be included into the shell. This editor should also include a component which can define backtracking rules to allow the system to undo previous allocations when certain conditions arise. The user interface in the ACT system could be extended further to include a facility that enables the user to relax some of the constraints when required. Future work on the ACT system would essentially be to convert it into an expert system shell for solving a more general school timetabling problems and to more truly reflect the expertise of the timetabler in creating an acceptable timetable.

## 8.6 Conclusions

The objective of designing and implementing a timetabling system is to enable the assignment of lessons to timeslots while satisfying all the necessary constraints. The timetable generated should be complete, must be of acceptable quality and produced in a reasonable time so that it can be of practical use. An in depth study of the development of a timetabling system using a hybrid of heuristics and an expert system to assist the timetablers of high schools was undertaken. From this study it is possible to draw some tentative conclusions concerning the appropriateness of expert systems in timetable construction for high schools, and other timetabling domains in general.

The timetabling approach using tabu search and simulated annealing has proved to be successful in achieving an acceptable automatic timetable within a short time. Rule based expert systems have further enhanced the search for a better solution in the area of satisfying soft requirements. The benefits of an expert system approach have shown the potential of expanding this work into a complete expert system

#### Summary and Conclusions

shell for solving high school timetabling problems. In conclusion, this contribution to the development of high school timetabling methods, using a hybrid heuristic and expert systems approach has brought a successful result. This is achieved by satisfying the hard and soft constraints for various schools. The heuristic models were able to produce a timetable very quickly and the rule based expert system managed to satisfy most if not all the soft requirements.

Finally it is appropriate to highlight several limitations of the research study. One of the main difficulties was in obtaining timetablers who were willing to be involved in this research with a sufficient level of commitment. Consequently, the significant limitation of the research is that it involves only three case studies. Therefore any conclusions drawn should only be regarded as a general observations from what experience suggests is a fairly typical timetabling situation. Secondly, the schools studied have relatively simple timetable involving a 5 day weekly cycle with a fully specified curriculum  $\omega$  the no subject options.

A comprehensive review of the timetabling literature has revealed several weaknesses with previous methods for solving timetabling problems. In particular previous methods have generally failed to take into account the very soft requirements prevalent in almost all timetabling situations. It is suggested, therefore, that a radically new approach for timetabling is required; and that the characteristics of expert systems suggest that they provide scope for further progress in this area.

## **BIBLIOGRAPHY**

- Aarts E., Korst J. and Laarhoven P. (1988), 'A Quantitative Analysis of The Simulated Annealing Algorithm: A Case Study for The Travelling Salesman Problem', Journal of Statistical Physics, Vol. 50, pp 189 - 206.
- Abramson D. (1991), 'Constructing School Timetable Using Simulated Annealing: Sequential and Parallel Algorithms', *Management Science*, Vol. 37(1), pp 98 - 113.
- Abramson D. and Abela J. (1992), 'A Parallel Genetic Algorithm Solving the School Timetabling Problem', Australian Computer Science Communications (Including Proc. Of The 15th. Australian Computer Science Conf), pp 29-31.
- Almond M., (1966), 'An Algorithm for Constructing University Timetable', *The* Computer Journal, Vol. 8, pp 331 - 342.
- Appleby J. S., Blake D. V. and Newman E. A. (1961), 'Techniques For Producing School Timetables On A Computer and Their Application to Other Scheduling Problems', *The Computer Journal*, Vol. 3, pp 237-245.
- Assad A. A. and Golden B. L. (1986), 'Expert Systems, Micro-Computers and Operations Research', Computers and Operations Research, Vol. 13(2), pp 301 321.
- Aubin Jean, Ferland and Jacques A., (1989), 'A Large Scale Timetabling Problem', Computers & Operations Research, Vol. 16(1), pp: 67-77.
- Barr A. and Feigenbaum E. A. (1983), 'The Handbook of Artificial Intelligence', Vol. 1, Pitman.

Barraclough E. D. (1965), 'The application of digital computer to the construction of timetables', *Computer Journal*. Vol. 8, pp 136-146.

Bonnet A. (1985), 'A.I. Promise and Performance', Prentice Hall.

Bryant N. (1988), 'Managing Expert System', John Wiley & Sons.

- Cangalovic, Mirjana, Schreuder, and Jan A. M. (1991), 'Exact Colouring Algorithm for Weighted Graphs Applied to Timetabling Problems with Lectures of Different Lengths', *EJOR*, Vol. 51(2), pp 248-258.
- Carter M.W. (1989), 'A Lagrangian Relaxation Approach to The Classroom Assignment Problem', *INFOR*, Vol. 27(2), pp 230-246.
- Chahal N. and de Werra D. (1989), 'An Interractive System For Constructing Timetables', EJOR, Vol. 40, pp 32 - 37.
- Chandrashekaran C. and Ramesh R. (1987), ' PROMOTER: AN Automated Promotion System', *Marketing Science*, Vol. 6(2), pp 23 - 34.
- Colorni A., Dorigo M. and Maniezzo V. (1991), 'Genetic Algorithms and Highly Constrained Problems: The Timetable Case', Lecture Notes in Computer Science, Vol. 496, pp 55-59.
- Cravo M. R. and Martins J. P., (1987), 'Automatic Generation of Timetable', Proceeding of EPIA-87, pp 226-245.
- Csima J. and Gotlieb C. C. (1965), 'Tests On A Computer Method For Constructing School Timetables', Communication of The ACM, Vol. 7 (3) pp 160-163.
- de Gans O. B. (1981), 'A Computer Timetabling System for Secondary Schools In The Netherlands', *EJOR*, Vol. 7, pp 175-182.
- de Werra D. (1971), 'Construction of School Timetable by Flow methods', INFOR - Canadian Journal of Operational Research and Information Processing, Vol. 9, pp 12-22.

- de Werra D. (1985), 'An Introduction to Timetabling', EJOR, Vol. 19(2), pp 151-162.
- de Werra D. (1989), 'An Interactive System for Constructing Timetables on a PC', EJOR, Vol. 40, pp 32-37.
- Dempster M. (1968), 'Two Algorithms for the Timetable problem. Combinatorial Mathematics and Its Applications', Proc. of Conf., Oxford, July 1969, pp 63 - 85.
- Dempster M. et. al., (1975), 'School Timetabling By Computer A Technical History', Education Research, pp 24-31
- Dhar V. and Ranganathan N. (1990), 'Integer Programming Vs Expert Systems: An Experimental Comparison', Communication of ACM, Vol. 33, pp 323 -336.
- Dinkel J. J, Mote J. and Venkataramanan M. A. (1989), 'An Efficient Decision Support System For Academic Course Scheduling', Operation Research, Vol. 37, pp 853-864.
- Doukidis G. I. and Paul, R. J. (1990), 'A Survey of the Application of Artificial Intelligence Techniques within the OR Society', JORS, Vol. 41(5), pp 363-375.
- Dowsland Kathryn A. (1990), 'A Timetabling Problem in Which Clashes Are Inevitable', JORS, Vol. 41 (10), pp 907-918.
- Dti (1990), 'Scheduling Interactive Activities: Corporate Meetings Planner', HMSO Publications, Case Study 11, UK.
- Duda R., Gashnig J. and Hart P. (1979), 'Model Design in the Prospector
  Consultant System for Mineral Exploration in Expert Systems in the Micro-Electronic Age', (Ed) Mitchie, D., Edinburg University Press.

- Eglese R. W. and Rand G. K., 'Conference Seminar Timetabling', JORS, Vol. 38 (7), pp 591-598.
- Eiselt H. A., and Laporte G. (1987), 'Combinatorial Optimisation Problems with Soft and Hard Requirements', JORS, Vol. 34 (6), pp 503-509.
- Evan S., Itai A. and Shamir A. (1976), 'On the Complexity of Timetabling and Multicommodity Flow Problems', SIAM Journal of Computing, Vol. 5 (4) pp 691-703.
- Falcao P., Pereira M., Ribeiro R. and Barahona P. (1990), 'Another Timetabling Approach', First International Congress on Expert Planning Systems, Brighton, pp 212-216.
- Ferland J. A. and Lavoie A. (1992), 'Exchanges Procedures for Timetabling Problem', Discrete Applied mathematics, Vol. 35, pp 237-235.
- Ferland J.A. and Roy S. (1985), 'Timetabling Problem for University as Assignment of Activities to Resources', Computer and Operations Research, Vol. 12(2), pp 207 - 218.
- Ferland Jacques A., Fleurent and Charles (1991), 'Computer Aided Scheduling for a Sport League', *INFOR*, Vol. 29 (1), pp 14-25.
- Fordyce K., Norden P. and Sullivan G. (1987), 'Review of Expert Systems for the Management Science Practitioner', *Interfaces*, Vol. 17 (2), pp 64 - 77.
- Glover F. (1977), 'Heuristics for Integer Programming Using Surroge Constraints', Decision Science, Vol. 8, pp 156-166.
- Glover F. (1987), 'Tabu Search', Technical Report, Center for Applied Artificial Intelligence, University of Colorado, Boulder,CO.
- Glover F. (1989), 'New Approaches for Heuristic search: A Bilateral Linkage with Artificial Intelligence', *EJOR*, Vol. 39, pp 119-130.

- Glover F. and MacMillan (1986), 'The General Employee Scheduling Problem: An Integration of Management Science and Artificial Intelligence', Computers and Operations Research, Vol. 13(5), pp 563 - 593.
- Golberg D. E. (1990), 'Genetic Algorithm in Search, Optimisation and Machine Learning', Addison-Wesley.
- Gotlieb C. (1962), 'The Construction of Class-Teacher Timetables', In Proceeding of the IFIP Congress 62, Vol. 62, pp 73-77.
- Grant T. J. (1986), 'Lessons for OR from AI: A Scheduling Case Study', *JORS*, Vol. 37 (1), pp 41 57.
- Griffith B. A. (1966), 'Remarks on A Computer Program for Construction of School Timetables', Communication ACM, Vol. 9, pp 35-43.
- Gudes E., Kuflik T., and Meisels A. (1990), 'An Expert Systems Based Methodology for solving resource allocation problems', In Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Charleston 3 Proc, pp 309 - 317.
- Gunasena U., Kumara S. R. T. and Soyster A. L. (1992), 'A Knowledge-Based System for Course Scheduling', Applied Artificial Intelligence, Vol. 3, pp 463-482.
- Harmon P. and King D. (1985), 'Expert System: Artificial Intelligence in Business', John Wiley & Sons, New York.
- Harmon P., Maus R. and Morrissey W. (1988), 'Expert Systems: Tools and Applications', John Wiley & Sons, New York.
- Hertz A. (1991), 'Tabu Search For Large Scale Timetabling', EJOR, Vol. 54, pp 39-47.
- Hertz A. (1992), 'Finding A Feasible Course Schedule Using Tabu Search', Discrete Applied Mathematics, Vol. 35, pp 255-270.

- Hertz A. and de Werra D. (1987), 'Using Tabu Search Techniques for Graph Colouring', Computing, Vol. 39, pp 345-351.
- Hilton A. J. W. (1980), 'The Reconstruction of Latin Square With Application to School Timetabling and Experimental Design', Mathematical Programming Study, Vol. 13, pp 68-77.
- Jackson P. (1986), 'Introduction to Expert Systems', Addison-Wesley, Wokingham, England.
- Johnson D. (1990), 'Timetabling University Examinations', JORS, Vol. 41 (1), pp 39 47.
- Johnson D. (1993), 'A Database Approach to Course Timetabling', JORS, Vol. 44 (5), pp 425-433.
- Johnson D. and Ismail Z. (1992), 'An Application of Tabu Search and Simulated Annealing in Solving a High School Timetabling Problem', Management Research Series, LUBS, Series 22.

Johnson K. (1980), 'Timetabling', Hutchinson publication.

- Johnston H. C. and Wolfenden K. (1969), 'Computer Aided Construction of School Timetables', *Information Processing*, Vol. 68, Proceeding of the IFIP Congress 1968, pp 1368-1376.
- Juginger W. (1986), 'Timetabling in Germany A Survey', Interface, Vol. 4 pp 66 -74.
- Kang L, Vonschoe G. H. and White G. M. (1991), 'Complete University Timetabling Using Logic', Computers and Education, Vol. 17 (2), pp 145 -153.
- Kang Le, White and George M. (1992), 'A Logic Approach to the Resolution of Constraints in Timetabling', *EJOR*, Vol. 61 (3), pp 306-317.

Kastner J. K. and Hong S. J. (1984), 'A Review of Expert System', *EJOR*, Vol. 18 (3), pp 285 - 292.

Keller R. (1987), 'Expert System technology', Yourdon Press, Prentice Hall.

- Kiaer L. and Yellen J. (1992), 'Weighted Graphs and University Course Timetabling', Computer and Operations Research, Vol. 19 (1), pp 59-67.
- Kirkpatrick, S. Gelatt Jr., and Vecchi M. P. (1983), 'Optimisation by Simulated Annealing', Science, Vol. 220, pp 671-680.
- Kitagawa F and Ikeda H. (1988), 'An Existential Problem of A Weight-Controlled Subset and Its Application to School Timetabling Construction', Discrete mathematics, Vol. 72, pp 195-211.
- Konya I., Somogyi P. and Szabados (1978), 'A Method of Timetable Construction by Computer', Period. Polytech. Elect. Eng., Vol. 22, pp 171 -181.
- Kovacic M. (1993), 'Timetable Construction with Markovian Neural Network', EJOR, Vol. 69, pp 92 - 96.
- Land F. and Doukidis G. (1988), 'In Seaarch of the Expert System Product', Expert Systems for Information Management.
- Laporte G. and Desroches S. (1986), 'The Problem of Assigning Students to Course Sections in Large Engineering School', Computer and Operation Research, Vol. 13, pp 387-394.
- Lawrie N. (1968), 'School timetabling by computer', In Aspects of Educational Technology, by Dunn W. and Holroyd C., Vol. 2, pp 619 634.
- Lawrie N. L and Veitch H. (1975), 'Timetabling and Organisation in Secondary Schools', CEER Publishing Company, New Jersey.
- Lawrie N. L. (1969), 'An Integer Programming Model of A School Timetabling Problem', Computer Journal, Vol. 12, pp 23-46.

- Leonard Barton D. and Sviokla J. J. (1988), 'Putting Expert Systems to Work', Harvard Business Review, Mar-April, pp 91-98.
- Lewis C. F. (1961), 'The School Timetable', Cambridge: Cambridge University Press.
- Lion J. (1966), 'Matrix Reduction Using the Hungarian Method For The Generation of School Timetables', Communications of The ACM, Vol. 9 (5), pp 349-354.
- Lion J. (1967), 'The Ontario School Scheduling Program', *The Computer Journal*, Vol. 10, pp 14-21.
- Loo E. H., Goh T. N. and Ong H. L. (1986), 'A Heuristics Approach to Scheduling University Timetable', *Computers and Education*, Vol. 10, pp 379-388.
- Lotfi V. and Cerveny R. (1991), 'A Final-Exam Scheduling Package', JORS, Vol. 42 (3), pp 205 - 216.
- Lundy M. and Mees A. (1986), 'Convergence of an Annealing Algorithm', Mathematical Programming, Vol. 34, pp 111-124.
- Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., and Teller E. (1953), 'Equations of State Calculations by Fast Computing Machine', Journal of Chemical Physics, Vol. 21, pp 1087 - 1092.
- Minsky M. (1968), 'Matter, Mind and Models', in Sematic Information Processing, MIT Press.
- Monfroglio A. (1988), 'Timetabling Through A Deductive DataBase: A Case Study', Data and Knowledge Engineering, Vol. 3, pp 1-27.

Mulvey J. M. (1982), 'A Class/Time Assignment Model', EJOR, Vol. 9, pp 64-70.

Murphy C. J. (1987), 'Timelog - An Intelligent Spreadsheet For School Timetabling', SIGART Newsletter, Vol. 101, pp 21-23.

- Nisenbaum P. (1994), 'Building Flexible User Interfaces', Information Systems Managements, Winter Edition, pp 62 - 67.
- Oxford System Groups (1970), 'School Timetables by Computers and Programs for Preparing School Timetables by Computer', Oxford.
- Papoulias D. B. (1980), 'The Assignment-To-Day Problems in A School Timetable A Heuristic Approach', EJOR, Vol. 4, pp 31-41.
- Reeves C. (1993), 'Improving The Efficiency of Tabu Search For Machine Sequencing Problems', JORS, Vol. 44(4), pp 375 383.
- Rich E. (1983), 'Artificial Intelligence', McGraw Hill
- Salazar A. and Oakford R. V. (1974), 'A Graph Formulation of a School Scheduling Algorithm', Communications of the ACM, Vol. 17, pp 696-698.
- Schmidt G. and Strohlein T. (1979), 'Timetable Construction An Annotated Bibliography', *The Computer Journal*, Vol. 23 (4), pp 307-316.
- Selim S. M. (1988), 'Split vertices in Vertex Colouring and Their Application in Developing a Solution to The Faculty Timetable Problem', *The Computer Journal*, Vol. 31 (1), pp 76-82.
- Sell P. (1985), 'Expert System A Practical Introduction', MacMillan, London.
- Shirai Y. and Tsujii J. (1984), 'Artificial Intelligence: Concepts, Techniques and Applications', John Wiley, 1984.
- Shortcliffe E. H. (1976), 'Computer Based Medical Consultants: MYCIN', Elsevier, New York.
- Silver E. A., Vidal R. V. V. and de Werra D (1980), 'A Tutorial on Heuristic Methods', EJOR, Vol. 5, pp 153-162.
- Slagle J. R. and Hamburger H. (1985), 'An Expert System For A Resource Allocation Problem', Communication of the ACM, Vol. 28 (9), pp 34 43.

- Smith G. (1975), 'On Maintainance of The Opportunity List for Class-Teacher Timetabling Problems', Communication ACM, Vol. 18, pp 203-208.
- Tripathy A. (1984), 'School Timetabling A Case in Large Binary Integer Linear Programming', *Management Science*, Vol. 30 (12), pp 1473-1489.
- Tripathy A. (1992), 'Computerised decision Aid for Timetabling A case Analysis', Discrete Applied Mathematics, Vol. 35, pp 313-323.
- Verbraeck A. (1990), 'A Decision Support System For Timetable Construction', First International Conference on Expert Planning Systems, Brighton, England. pp 207-211.
- Vlagalovic M. and Schreuder J. A. M. (1991), 'Exact Colouring Algorithm for Weighted Graphs Applied to Timetabling Problems With Lectures of Different Lengths', EJOR', Vol. 51, pp 248 - 258.
- Walters J. and Nielsen J. (1988), 'Crafting Knowledge-Based Systems', John Wiley, New York
- Welsh D. J. A and Powell M. B.(1967), 'An Upper Bound for The Chromatic Number of A Graph And Its Application to Timetabling Problems', The Computer Journal, Vol. 10, pp 85-86.
- White G. and Wong S. (1988), 'Interactive Timetabling in Universities', Computers and Education, Vol. 12 (4), pp 521 - 529.
- Winfield M. J. (1982), 'Expert Systems: An Introduction for the Layman', Computer Bulletin, pp 6-7 and 18.
- Wood D. C. (1969), 'A Technique for Colouring a Graph Applicable to Large Scale Timetabling Problems', *The Computer Journal*, Vol. 12, pp 317-319.
- Yule A. P. (1967), 'Extensions to the heuristic algorithm for university timetables', Computer Journal, Vol. 10, pp 360-364.

# Appendix A

Appendix A contains a timetable from Limehurst High School which was produced manually.

22/3	<u> </u>	MÇ	<u>и</u> г	<u> </u>	, ;	T.	<u>ų e</u>	S D	<u>A.Y.</u>		<u>+-₩_</u> E	<u>, p. t</u>	<u>E</u>	<u>.p</u>	<u>, y</u>		т.н.	U,R,	S.D.	<u>A Y</u>		FR	<u>i d</u>	Ay	
9A		4	3	4	2	-	4	3	4	2	<u> -</u>	12	3	4		<u> </u>	<u> </u>	د اد	4		-	2.	3	4	
Gali	Ew <sup>16</sup>	м 5 <sup>2</sup>	Gall	Ma	9 Gh	Fi <sup>6</sup>	Sw 1	DS	Ga <sup>11</sup> 8	м 5 <sup>2</sup>	Вđ	вa	Al	Gh <sup>9</sup>	Ì	16 Ew	Gall		Вđ	G11 G1	DS	DS	LC Al	M.5 <sup>2</sup>	16 E#
Gr <sup>Mus</sup>	Ma <sup>15</sup> 13	SAS	Mi	DS	Gy7	Gh 7	Gr Ja	WO Ma	$\frac{Mi}{Pa}$	SAŜ 3	Ha	На Sh	Gr	Gy′ Ja <sup>6</sup>		ма́₹ 13	G <sup>Mu</sup> l	Gru	No Sh	Mi pl2	Sc	Sc	Gr	SAS	13 15
Pa <sup>12</sup>	SJS'	MATE	Pa <sup>1</sup> 9 ENC	WO SCI	Ја" НПМ	Gy НUМ	Pi EXP	SCI	ENG	Am MAT	TW		Ja EXP	ним	INT	SJS Lan		12 12	Tw DEs	ENG	Ba	Ba	JA EXP	Am MAT	SJ 5 LANC
9B	Ga <sup>11</sup>	Fi <sup>6</sup>	Ew14	rc	Ms 4	LC/ Sw	15 15	Ms <sup>2</sup>	f17	Gall	Iŝ	DS	Bđ	A1		Ms	BdT	Bd	DS	DS	Bd	Bd WO	DS	Gal	Fi
BdLT	Mi <sup>8</sup>	Gy <sup>7</sup>	Is Ma	Al Gr	SAS	Al Gr	ма́ Є 13	4 SAS	Gy'	M1 <sup>6</sup>	Mat	Ma	Ha Sh	Gr		SÄS 3	Fi	Fi	Ma Bu	Ma B11	Sh	Sh	Sc Wo	Mi' 12	Gh GV
Fi <sup>b</sup> SJS <sup>13</sup>	Pa <sup>12</sup>	Ja <sup>9</sup>	SJS-	Pi	Am .	Ja Bio	SJS LANG	And Marti	ним	ENG	SJS	WO SCT	Tw DES	Pi	INT	мат	sjs,	sts.	SCT	SCT	Tw	TW	SCI	ENG	HUM
	5-		PAUL .			1205	<del></del>	10010			2		- 11	<u>-rxp</u>	-		10.				<u></u>			1.07	
8A Cr <sup>10</sup>	Co <sup>-</sup>	LC Al	Ma DS	1s <sup>14</sup> 15	Ga <sup>11</sup>	Bd Bu	Bđ Bu	Is14	DS` Ma	Co <sup>5</sup>	Ms SAS	A1	Gå⊥ ⊶8	Mar		Bđ	Cr	C <sub>1</sub> 0	CO CO	M3	Co <sup>5</sup>	Ga 11	14 15 15	A1 SV	M-13
Sm <sup>4</sup>	Gh <sup>9</sup>	Gr Pi	WO	Mat <sup>-</sup> SJS <sup>13</sup>	<sup>M1</sup> 12 Pa	Sh Tw	Sh Tw	Ma 13	Wo	Gh	6mA	GF Ja	Pa <sup>12</sup>	Bu		Sh	Sm <sup>®</sup> Ch <sup>9</sup>	5m 9 Ch	Gh	Am <sup>3</sup>	F19 Gh	M1 12 Pa	Māf sj3	Gr Ja	SAS 3
Gh <sup>9</sup>	BUM	EXP	SCI	LANG	ENG	DES	DES	LANG	SCL	ним	MAT	EXP	ENG	SCI	TNT	Tw Des	TUTO		ним	MAT	ним	ENG	LAN	EXP	MAT
8 <u>B</u>	MB <sup>2</sup>	Ew <sup>10</sup>	fi <sup>6</sup>	Bđ	Bđ	DS	DS	A1	16 EW	Ba	DS	M <b>s<sup>2</sup></b>	Fi <sup>6</sup>			LC Al	Tw <sup>2</sup>	Tw2	Ga 	Fi <sup>o</sup>	Ga <sup>11</sup> 8	6 19	M 5 <sup>2</sup>	EW 15	₩¥₩ Al
Tw⁻ Mi₿	SAS <sup>4</sup> Am3	Mat s13	Gh y	Bu Sh	BU Sh	Wo	Ma Wo	Gr Jo	Māť 513	Sh	WO	Am <sup>3</sup>	Gh <sup>-</sup> Gv <sup>-</sup> 7	Mĭ _12		Gr Pi	Mi <sup>o</sup>	Mi		G7 G7	M1 Pa <sup>12</sup>	GN 7	SAS Am	Māt sjs	Gr Ja
Sh	MATH	LANG	ру НИМ	TW		SCI	scì	EXP	LAN	Tw DES	SCI	мат	HUM	PA ENG	İNT	EXP	TUT		ENG	HUM	ENG	ey RUM	алт I	TAN	Pi FYP
7.	·		2			.2		C-5		16	5	5		_16		Co5	5!	5	16	-	M-4		101	<u>1000</u>	D.J
<u>-7A</u>	Ma	Ma	MS	C06	LC/	M'S A	°9	<b>5</b>	AT	EQT	Co	Co <sup>-</sup>	DS	EW 19		F 1 6	A1	AI	EW	Bđ		A1	6	Ba Ba	Bu
Al Am3	Sc Wo	SC Wo	SAS Am 3	F1 Gh <sup>9</sup>	Al Fu	SAS Am3	Gh Gy <sup>7</sup>	ст Gy <sup>7</sup>	Gr Pi	MAE	Fi <sup>0</sup> Ch <sup>9</sup>	Fig Gh <sup>9</sup>	Ma Wo	Mat 535		Gh <sup>9</sup>	Am- 		Mat 13	WO	Am3	Gr	Fi Gh <sup>9</sup>	Sh	Sh
Ma <sup>15</sup>			İ	Gy <sup>7</sup>	Gr Pi	MAT	-				Gy <sup>7</sup>	Gy <sup>7</sup>	SCT		INT	Gy '	TUT		<b>Š</b> JS	Sh Tw		Pi	Gy <sup>7</sup>	Tw	Tw
•		SCI	MAT	2NL	EXP		Или	ENG	EXP	ANG	E N(	In	301	LAN	i	1 m	$\leq$	$\geq$	LANG	DES	MAJ	EXP	inns	230	220
<u>7</u> B	вđ	вđ	LC	Ga <sup>1</sup>	DS	-16 -15	Ga <sup>1</sup> 8	$Ga^{11}$	Ms4	LC/ Sw	Ga 1 12		E	Bđ		$G_8^{11}$	EW 7	E96 7	2 M's	EW	Al	Ms _1	6 <sup>11</sup> 6 <sup>8</sup> 8	DS	DS
Ew <sup>10</sup>	Bu Sh	Bu ISh	Gr	Pa <sup>1</sup>	Ma Wo	Mat 13	Pa	Mĭ Pal2	SAS Am 3	Al Gr	Pã 16	Pā* .8	Mat 13	Wo		M12 Pa	DS'	DS _14	SAS	ма 13	Gr	SAŠ Am3	M1 Ew <sup>16</sup>	SC	ŝ
Is <sup>14</sup>	Tw	Tw	Pi	HUM, ENG		6JS	Ew <sup>11</sup> HIIM	Ew <sup>16</sup>		Ja Pi	ĒŴ	YI HUM	SJS	5n Tw	- C	Ним	IŜ	IS	Am	SJS	Ň		ним/	Wo	Wo
(	DES	DES	EXF		SCI	LAN	ENG	ENG	MATH	EXF	ENG	ENG	LANC	•	INT	ENG	TUT	OR	MATH	LAN	EXP	MAT	ENG.	SCI	SCI

A manually produced timetable for Limehurst High School

157



ğ. ..

# Appendix B

Appendix B, contains a timetable from Market Bosworth High School which was produced manually.

7 I P	-	1993/94						
		ΥΣΑΝ 7 Ν 8	т <u>ен</u> а Н	YEAR 9 M B				
		1 DECITIN	SELENCE COENCIN	11년 1년 18년 18년 18년 18년 18년 18년 18년 18년 1				
		2 MAN PS HO SHA - 2 - 74 CW	MATHS SECOND SECOND					
	идираў	3 CS PC JW AW BS IS CM AR	5244.152 25 52 25 12 00 FM FM 53 FO 36 56 VM 136 CM MISTORY 4	TI BH PB AP				
			GEGENERNY EN REENGLICH	SCIENCE ON RE BU ONA				
		S PH PH 3F ES IT BH PB CO		FRANCH SCIEWLE				
		tt aw pe CR ES FH SF OK	GB FQ VH CS PC JW AN	8F 8D JC AB BS IG CH 4R				
		A DE JC AB KI TT BH PB CR		ESENCIA CONCESSO 25 16 CE CE CA CA CO				
		2 DS IG CH AR FM FM CF DK	CESIGN SH4 R5 6B SH4 SCIENCE B0 CS PC 70 40	GAMEC CP 45 HE MU				
A n	TUFSDAY	3 54 He 38 FA OFSIGN	HATTORY & STORAPHY AR ENGLISH	MAINS FRENCH				
ıanu	, service	4 25 PG H9 SH0 R9 28 SH4	BEIG CH CE TT SH PR CR	ENGLISH SCIENCE				
ally <sub>l</sub>		S CS PC JH AN SP JC AR XI	ENGLISH FREMEN	140 141 C2 UA				
prod		• FH RE ENGLISH GATES	SCIENCE MOTHS CS PC JW AN BF BO JC BO	OESIGN      HISTORY E        HISTORY E      GEOGRAPHY        SH6 RS 88 SH6      GEOGRAPHY        CR      TT 84 PB KP				
ncec		HISTORY &						
l tim		1 TT BM PB KP CS PC JM AN	PE DE MU SE LE VIU FO GE DESJSH	ENGLISH MAIHS				
etab	WEDNESDAY.	2 FM FH ES DA 85 IG CM AR	- 76 G8 VH CR	SCIENCE GEOGRAPHY C3 PC JW AN TT SW P8 KP				
le fo		3 45 16 CH AR 38 0X 13 110	EN RE ENGLISH MATHS	HISTORY S GEGGAAPHY SCIENCE TT BW PB XP CS PC JN AM				
r Ma		4 SCIENCE MAINS CS PC JN AN SF JC AT XI	ERENCH HIS/GEO HIS/GEO FRENCH	<u>РЕ НЕ РЕ DESIGN</u> 50 VN SH, РЗ 88 SH., ES 18 HB BO				
ırket		S GEOGRAPH	C SCIENCE ENGLISH BE	PE MU OR FREMCH SG VN HB GB FQ 85 1G CH AR				
Bosy		6 5% RS 88 5% FM FH E3 0	N NATHE SCIENCE AF 80 JC AT CS PC JM AM	FRENCH GAMES 85 IG CH-CE SG IB VH MB				
vortl		1-270R PERIOD	'UIGR PLETA	титон <u>, исл</u> ісо				
h Hig		HISTORY 3 GEORAPHY PE OR	ч натия лесиси "	T <u>CIEN</u> CE ENGLISH				
sh So			BENCH ENGLISH	08 25 HE MU DECOREPHY				
hool	<u>Invriday</u>	Г СС 18 ИН НО ВР ЈС АТ КІ РЛЕНЦИ <u>Science</u> • <u>Rcience</u> реди	H AL ENGLISH ON ON TREAT	ГОЈ ESI CS GB ТТ ОН РВ КР CESIGN НАТИЗ				
		S MATHS ENGLISH	E PH SP FQ GB PV HB	CR OF CO JC AT				
		EF JC AT KI FR FR 3F	QERIGH      54 18 VH rol ce        V      Stars and an and an and an and an and an and an and an and an and an and an and an and an and an and an an an an an an an an an an an an an	GANES OF CS PC JU AN GANES OF GERMAN HE MU QE BE				
			BD CS PC JN AM	54 (8 VM 14 ) 85   68   FQ   ES				
		L BS IG CH AR CS PC JW AW	ULSIORY & CALINS GEOGRAPHY CALINS TT DH PE CR BF BD JC AB	ENGLISH RE DR PE FM FH 37 DV ES FQ CG IB VH				
		2 TT BW SE CR BF BO JC AB	ERENCH GAMES	HE THE DE TE ENGLISH				
		3 AL ENGLISH	GAMES HISIGHT & GEOGRAPHY SQ 18 VH HB TT SN PB CR	ESENCH NATHS es ig ch ce ef ed jc ag				
	-ridat	4 18 HB FG GB SHO R5 88 SH	HATHS ENGLISH BE EN	ICIINCE FRENCH CS PC JW AM BS IG CH AR				
		5 PG G8 5G VH FN C3 5F	15 <sup>21</sup> <u>9CIENCE</u> H CS PC JN AM	HATHS HISTORY S GEOGRAPHY EF BO JC AB TT BH PB KP				
١		1011281 10111111111111111111111111111111	ENGLICH RE SHO AS 90 SHO	HISTORY - GEOGRAPHY SE MU HE				

160

62

FH FH 30

3Ho 45 98 5Ho CA

<u>ни не</u> 68 рс

SG IO VH

17 98 08 XP

or 30 48 KI

85 10 CH AR

## Appendix C

Appendix C, contains a timetable from Stonehill High School which was produced manually.

. ·	MONDAY							WEDNESDAY THURSDAY							FRID					
	12	3.4	56	79	12	3 +	5 G	78	1 2	J 4	5 G	73	12	34	5 6	78	1 2	3 4	ςG	78
9 S.X 9 S.X	MATHS	Р Ē   	MUS FOR	HISTOUEX	ёнgush		GERMAN FRENCH	DTA	MA ENG	DTA	SCIENCE	FRENCH PSE MUS RE PSE HUS RE	SCIENCE	₽E   	MATHS	HISTAGEOC	GER FR FR RE FR MUS FR FSE	ENGLISH   !	HISTIGEOG	DTA
4 *** 9 PL 9 CS	Енбызн Ј	FRENCH FR RE MUS FR	HIST2 GEXG	DTA	MATHS.	DTA	₽ E     	SCIENCE	H+ST\$46X4	P E	ENG MA	German French	HUT LGEXY	SCIENCE	FR MUS FR PSE FR RE	ENGUSH	PSE GER RE GER MUS FR RSE FR	MATHS	DTA	SCIENCE
305 305 205 205 205 205 205 205 205 205 205 2	MUS RE FRENCH	DTA HISTZ GEOG	MATHS FINGLISH	P E	SCIEXCE	ENGLISH	HSTLGEOG JCIENCE	GERMAN FRENCH V HISTEGBO	EKG MA	SCIENCE	FRENCH RE MUS PSE RE MUS PSE	DTA MA ENG	MATHS DTA	HUSTZGEGG FR HUS	PE	DTA	HISTEGEOG DTA	GER FR FR NSE FR NS FR RE FR RE FR GER	ENGROH	ENGLISH
8MC 3Fn			.	FR PJE MUS FR							French			FR RE FR PSE			V	MUS FR RE FR		
715 7h⊾ 7rg 7sk:	рта 	матня	PE	SCIENCE	HST 5 GEX4	REPSE FRENCH FRENCH MUSIC		ENGLISH	FRENCH MUSIC FRENCH RE ISE	MAERO	T DTA	SCIENCE	PE	MATHS	HISTLGEOG	FRENCH RE PSE MUSIC FRENCH	ENGLISH	\$CIBKŒ	MUSIC FRENCH RE FSE FRENCH	HISTLGEOG
714 714 714	РЕ   У	SCIENCE	DTA	ENGLISH	FRENCH	INSTACEOU	A man			RE PSE MUSIC FRENCH			MUSIC RE PSE PSE RE		ENGLISH	SCIENCE	SCIENCE	HISTSCOOL	MATINS	FRENCH

• •

1994-5

...

.

# Appendix D

This paper is published in the Loughborough University Management Series, Paper No. 22, 1992.

## ABSTRACT

A number of applications of Tabu search have been reported since the idea was originally proposed by Glover in 1989. In this paper we consider the use of mainly Tabu Search in solving a typical school timetabling problem where each student follows a defined curriculum. The paper examines the alternative use of simulated annealing for one aspect of timetable development. This enables a simple comparison of the two approaches to be made.

Keywords: Timetabling; Tabu Search; Simulated Annealing

#### INTRODUCTION

A recurring problem in virtually all educational establishments is the creation of a suitable timetable on at least an annual and very often a term by term basis. This problem becomes generally more difficult as the size of the institution and the flexibility of the curriculum increase. At one extreme in a small junior school where each class is taught exclusively by one teacher, the timetable is simply a matter of allocating each period during the week to a particular subject in accordance with defined curriculum requirements. This may well be left entirely to the discretion of the teacher concerned. Alternatively, in a sixth form college or higher education institution, timetables have to allow for large numbers of students to select subjects/courses on a modular basis to make up increasingly varied 'A' level or degree programmes. Whilst there is no guarantee that any desired combination of subjects/courses will necessarily be feasible, the job of the timetabler is to allow as many choices as possible to be satisfied.

In a review article, De Werra<sup>1</sup> noted that there were two distinct stages to the timetabling process, namely:

- 1. "First, the curricula are defined for each class or for each group of students," and the necessary teaching resources (staff, equipment and space) are assigned.
- 2. "Second, when an agreement has been reached concerning these assignments of resources, then one tries to see whether a workable detailed timetable can be worked out which is compatible with all the previously defined requirements."

For most senior schools, the first stage is not really the responsibility of the timetabler. Curriculum requirements are either pre-specified by some "authority" or at least decided on at a senior level within the school. The major stage 1 decision is then which teacher shall be assigned to each class for each subject, and this is usually a departmental decision rather than a

File:paper

timetabling one. Once a particular assignment of teachers has been decided, this will effectively determine the space (ie room) allocation as it can usually be assumed that there will be a room available for most, if not all, teachers for each period of the week.

The traditional school timetabling problem is therefore usually identified with stage 2, that of ordering the various classes or groups over a specified period of time (usually a week) subject to various restrictions on the resources involved. These restrictions are principally that no student should be taking more than one subject in the same period, and that no teacher should have more than one class at the same time. These are usually termed the 'hard' constraints; in contrast to a variety of 'soft' constraints (or desirabilities) such as that a teacher's free periods should be spread evenly throughout the week, and a class should not have multiple lessons in the same subject on any day unless these are deliberately planned as double periods. Local conditions will often give rise to additional constraints, either hard or soft, particularly when part-time teachers or outside facilities are used. In essence, therefore, the problem is to allocate each period of the week to a particular subject (and by implication a particular teacher) for each class in accordance with pre-specified curriculum requirements, so as to satisfy all of the hard constraints and as many of the soft constraints as possible.

In its basic form involving just the hard constraints indicated above, the problem can be conceptualised in graph theory terms or alternatively as a set of linear constraints relating to a series of binary variables  $x_{ijk}$  where  $x_{ijk} = 1$  if class i takes subject j in period k, and  $x_{ijk} = 0$ otherwise. These two approaches are summarised in Johnson<sup>2</sup>.

#### A PARTICULAR EXAMPLE

To illustrate a typical school timetabling problem, consider the case of a local high school taking pupils from 11-14 years in three separate year groups (years 7, 8 and 9). Each year
File:paper

group is divided into 2 sections or bands (A and B) and within each band there are typically 4 forms or classes. The whole school therefore comprises about 650 pupils in 24 forms.

The teaching week is divided into 30 periods (6 periods per day) and each period is devoted to one of 12 distinct subjects, with no free periods. Each year group follows a defined curriculum with typically 4 or 5 periods devoted to major subjects such as English or Science, down to just 1 period a week for minor subjects such as Music.

The school's staffing is such that in most departments there are just enough teachers to teach each of the four forms in one year band at the same time. Thus the timetable is basically planned on a 'band' basis rather than a 'form' basis and for most subjects, all the pupils in a particular band will be taking the same subject in any period. The only exception to this general rule is in the case of departments such as Music or Drama where there is only 1 teacher available. In these cases, that subject is subsumed within one of the major subjects such as English and one of the four forms in any band is taught Music (say) rather than English on a rotational basis. Typically, therefore, 5 periods might be devoted to English/Music in a particular year band, and these would be timetabled as follows:

Form	Period 1	Period 2	Period 3	Period 4	Period 5		
1	English	Music	English	English	English		
2	English	English	Music	English	English		
3	English	English	English	Music	English		
4	English	English	English	English	Music		

Timetabling is the responsibility of a senior teacher (the Head of Science) and he does this manually each term starting from the structure of the previous term's timetable. However, occasional changes to the curriculum and more frequent changes of staffing (particularly the use of part-time teachers) mean that the timetable can seldom be carried over, and very often

the changes are such that he has to virtually redesign the whole timetable as only a limited amount of 'tinkering' can be done before the timetable begins to fall apart.

### TABU SEARCH AND SIMULATED ANNEALING

A number of alternative approaches to combinatorial optimisation, notably tabu search (TS), simulated annealing (SA) and genetic algorithms (GA) have been developed in recent years and various recent papers have compared the efficiency of these techniques for different problem situations. Results seem to indicate that in most situations, TS performs as well if not better than other methods.

Tabu search was originally proposed by Glover<sup>3</sup> and, in the words of Glover and Laguna<sup>4</sup>, is "designed to cross boundaries of feasibility and local optimality". Tabu search is basically an iterative procedure which starts from an initial feasible solution and strives towards a globally optimum solution by a series of stepwise moves through a series of 'neighbourhood' points. At each step, a set  $V^*$  of points in the neighbourhood N(s) of the current solution s is generated, and a move is made to the best solution s' in  $V^*$ . This process of neighbourhood search is embodied in a number of other techniques, such as descent methods and simulated annealing. The feature that characterises tabu search, however, is the use of a selective memory of previous solutions which modifies the neighbourhood N(s) by excluding certain elements that are deemed to be 'tabu'. The tabu list can be determined in a variety of ways, but the most frequently used criterion is one of 'recency' where tabu elements are the most recent so - many solutions which possess particular characteristics or attributes.

In contrast, simulated annealing is a more haphazard technique whereby a neighbourhood point (s') is chosen randomly rather than optimally. If this new point improves the objective, the move is accepted and s' becomes the new solution. If s' does not improve the objective it may nevertheless be accepted with probability  $e^{-d/T}$  where d is the associated change in the

objective and T is a control parameter (the 'temperature') which decreases monotonically with each iteration. With simulated annealing there is no guarantee that a move is made to the best point in the neighbourhood set.

In this paper we will apply both tabu search and simulated annealing to the timetabling problem described in the previous section. Other applications of tabu search and simulated annealing to particular timetabling situations have been described (see for example Hertz<sup>5</sup> and Abramson<sup>6</sup>), but none of these have compared the two approaches in the context of a realistic timetabling problem. The procedure described here has been programmed in Fortran 77 to run on a Hewlett Packard mainframe computer, although the CPU times incurred indicate that it could feasibly be run on a suitable micro-computer.

### SOLUTION PROCEDURE

The problem presented by the High School described above is essentially to timetable each of 6 year bands over a 30 period week in accordance with a curriculum requirement for up to 12 subjects with prescribed frequencies (numbers of periods) for each subject. In addition, staffing considerations dictate that any subject can only be taught to at most one year band during any period. It would also be desirable if the number of occasions when any subject is taught more than once (on the same day) to any year band is minimised, and teacher free periods are spread as evenly as possible over the week. These last two soft requirements are likely to be positively correlated in this instance, as spreading the lessons in any subject throughout the week will very probably lead to the same feature for the teacher free periods. For this reason, it was decided to concentrate on spreading the multiple lessons, and hope that this would simultaneously spread the free periods. In any event, the timetabler confirmed that he did not consciously consider teacher free periods when devising the timetable, and teachers had to "accept what came out".

The overall procedure then falls naturally into the following stages:



The starting solution achieved in Stage 1 will produce a timetable which has no subject taught more than once during any period. This will not be a workable timetable, however, as curriculum requirements have not been considered. Stage 2 will turn the Stage 1 timetable into a theoretically feasible timetable by satisfying all the hard constraints. Finally, Stage 3 will provide a realistic timetable from a student perspective. It may need further manual adjustment to incorporate special factors such as concentrating the workload of part-time teachers.

## Stage 0

This stage defines the problem in terms of:

- the number of year bands (b = 6)
- the number of subjects (n = 12)
- the number of periods per day (p = 6)
- the curriculum requirement matrix for year band i and subject j ie a<sub>ii</sub>

Because pupils are not allowed any free periods

$$\sum_{j} a_{ij} = 5p$$
 (i = 1,..., b)

## Stage 1

If each subject is denoted by an integer code in the range 1 to n, then this stage requires the generation of a random permutation of 6 different integers ( $\leq$ n) for each of the 5p periods in a week. This set of permutations gives a starting 'timetable' in terms of a subject r<sub>ik</sub> for year band i in period k. As already noted, this timetable will not necessarily satisfy the curriculum requirements unless

 $c_{ij} = a_{ij}$  (all i, j) where  $c_{ij} = \text{count}_j$  ( $r_{ik}$ ; k = 1,...., 5p) denotes the number of occurrences of subject j in row i of the matrix  $r_{ik}$ .

### Stage 2

To create a feasible timetable which does satisfy the curriculum requirements, it is necessary to consider the deviance matrix

$$D = (d_{ij}) = (c_{ij}) - (a_{ij})$$

which indicates the excess/shortfall in the number of periods of subject j timetabled for band i. Clearly the matrix D will be stochastic with

$$\sum_{j} d_{ij} = 0$$
 (i = 1,..., b)

For each band i, any subject with excess periods  $(d_{ij} > 0)$  must be replaced by another subject with a shortfall  $(d_{ij} < 0)$ . By a series of such 'swaps' the starting timetable  $r_{ik}$  will be transformed into a feasible timetable which satisfies the curriculum requirements. If we employ a tabu search approach, the neighbourhood set will be defined by those solutions which can be produced by simply swapping some element (subject) in  $r_{ik}$  which has a positive

File:paper

count in  $d_{ij}$  for some other subject. The tabu list will define which swaps are not legitimate and will be made up of all subjects which have a positive or zero count in  $d_{ij}$ , or which are already present in the same column of  $r_{ik}$  (ie are being taught to some other band during the period under consideration). While legitimate swaps are available, the subject with the largest positive count in the deviance matrix is chosen, and it is replaced by a subject chosen at random from those available to replace it.

Ultimately, however, as the timetable approaches feasibility, it is likely that no legitimate swaps will be possible before all the  $d_{ij}$  elements have become zero. If and when this occurs, the definition of the tabu list changes slightly and now no longer includes those subjects with a zero count in the appropriate row of the deviance matrix. The effect of this is to permit swaps with subjects whose curriculum requirement is exactly satisfied and thereby replace (or reduce) one excess subject by another. The procedure then reverts to the original definition of the tabu list and proceeds as before until no legitimate swaps are available, or until the deviance matrix has become zero.

Limited experimentation with different numbers of subjects and periods shows that the CPU time to produce a feasible solution increases more or less linearly with the number of subjects. In contrast, there is some evidence that the increase from a 7 to 8 period day takes proportionately longer to timetable than the increase from 6 to 7 periods per day. The results are displayed in Figure 1.

#### **INSERT FIGURE 1 HERE**

### Stage 3

Examination of the timetables produced by Stage 2 shows that there are frequent repetitions of subjects within the same day. Indeed, the major subjects which have 4 or 5 periods devoted to them could have as many as 4 periods in the same day. In some situations, there may be a requirement for double periods for some subjects such as Games, but at the school in question,

all periods were explicitly planned as single periods and it was felt desirable to have as few occasions as possible when the same subject was taught twice (or more) on the same day. Accordingly, Stage 3 of the timetabling programme is designed to minimise the occurrence of this.

It was decided that a simulated annealing approach would be appropriate with an objective function that would remove as much as possible of the repetition of subjects taught in any day. An obvious function that would achieve this is to maximise the number of different subjects taught in any day, over all 6 year bands and all 5 days. With no repetitions, the maximum possible value of the objective function is:

6 bands x 5 days x 6 periods = 180

A typical score for the timetables produced by Stage 2 is 25 per year band ie a total of 150. This implies an average of one repeated subject per day for each band, and this was considered generally unacceptable. It was felt intuitively that a score of at least 174 should be aimed for, implying an average of only 1 repeated subject per week for each band.

The first approach to achieving the desired spread of periods was to use simulated annealing. For this purpose the neighbourhood set for any timetable was defined as those new timetables that could be obtained by interchanging all the classes in any two periods on different days. This would clearly not violate any of the hard constraints previously satisfied, but would enable some (if not all) repetitions to be eliminated. The procedure is similar to one used in Johnson<sup>7</sup> for timetabling examinations. In that case, the objective was to minimise the number of occasions where students had to sit two examinations on the same day.

As described previously, simulated annealing is controlled by a temperature parameter (T) which decreases monotonically with each successive iteration. Lundy and Mees<sup>8</sup> suggest an iterative relationship for T of the following form:

 $T_i = T_{i-1}/(1 + g T_{i-1})$ 

where the constant g is chosen such that the temperature falls from some initial value  $T_0$  to a final value  $T_m$  in m iterations.

The procedure used was therefore as follows:

- (i) determine randomly two different days
- (ii) randomly select a period in each day
- (iii) interchange the sets of subjects taught in these two periods and calculate the change (d<sub>i</sub>) the objective function.
- (iv) determine  $T_i = T_{i-1}/(1 + g T_{i-1})$
- (v) if  $d_i > 0$ , accept the new timetable if  $d_i \le 0$ , accept the new timetable with probability exp  $(d_i/T_i)$
- (vi) repeat for  $i \rightarrow i+1$

The final timetable can then be chosen in various ways. A single best timetable is available as the one which yielded the highest score during the various iterations of the simulated annealing. Alternatively, if a number of high scoring timetables are produced, a selection of these can be retained so that the timetabler can choose the one most suitable for final modification and implementation.

The score obtained using simulated annealing at stage 3 is typically between 157 and 165. There is therefore a clear improvement to the spread of the multiple period subjects as a result of applying simulated annealing, but the timetable still falls some way short of the required score of 174 and typically has about two repeated subjects per week for each year band.

To try and produce a more acceptable timetable, a tabu search approach similar to that used in stage 2 was adopted at stage 3 in place of simulated annealing. In this case the neighbourhood set was defined by identifying any subject (A) having a repeat period in a particular day and exchanging that subject with some other subject (B) from a different day. Without any restriction on what could be exchanged, this process would almost certainly lead to a violation of the condition that each subject can only be taught to at most one year band in any period.

File:paper

The choice of subject B must therefore be constrained in two ways. Firstly, it cannot be in a period when subject A is already being taught to some other year band. Secondly, those subjects already being taught on the same day as A and all subjects being taught to other year bands in the same period as A, will be tabu. Where there are a number of possible exchanges that can be made, the one chosen is that which leads to the largest increase in the 'score' as defined above for simulated annealing.

This is illustrated by the following simplified example involving 3 days (with 5 periods per day), 6 subjects (A - F) and 4 year bands.

Year		Day 1				Day 2				Day 3					
Band	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1 2 3 4	A B C D	A D B F	B E F A	E A D B	D F A E	B C F E	D A E C	F B A D	E D C A	C E B F	C D A B	E A B C	F C D A	A B E F	B F C D
Tabu	colu	mns	on	days	2,3		*	*	*		*	*	*	*	

Tabu list for Day 1 Period 1: A, B, C, D, E Tabu list for Day 1 Period 2: A, B, D, E, F

The only case of a repeated subject is year band 1 on day 1 where subject A occurs twice. Subject A in either period 1 or period 2 must be exchanged with another subject in row 1 on either day 2 or day 3. Firstly, those columns on days 2 and 3 which already contain subject A are tabu (these are indicated by \*). Secondly, for both columns 1 and 2 there is a tabu list of those subjects already on day 1 or in the column in question. The two tabu lists are given below the table. It can be seen that there are no possible exchanges for the A in period 1, but the A in period 2 can be exchanged with the C in period 5 on day 2.

In practice, there will be cases where replacement of repeated subjects is impossible and the timetable produced will then give a score of less than 180. In repeated runs, however, the new

procedure with tabu search at stage 3 invariably produced scores in excess of 175 and quite often achieved the "perfect" score of 180. It would appear then that the use of tabu search procedures in this situation is more effective than simulated annealing, although it must be

File:paper

recognised that simulated annealing has only been used for the last stage in the process, that of 'fine tuning' an already feasible timetable in order to produce an even spread of subjects throughout the week.

### CONCLUSIONS

This paper has described how the techniques of tabu search and simulated annealing can be used at various stages in creating a timetable for a local high school. The situation concerned is not unduly complex as there are no elective elements to the timetable, but there are other special features which create local difficulties. In particular, there is a need to avoid teaching the same subject to more than one year band in any period, and if possible to produce a timetable which does not entail the students having a repeated subject on any day.

In company with other papers which have used both tabu search and simulated annealing to solve particular timetabling problems, the results seem to indicate that 'optimising' aspect of tabu search produces better results than the more 'random' nature of simulated annealing. In this case, a tabu search procedure invariably gives rise to a timetable which more effectively spreads out the multiple period subjects than does a method based on simulated annealing.

### REFERENCES

- 1. D. de Werra (1985) An introduction to timetabling. Eur. J. Opl. Res. 19, 151-162
- 2. D. G. Johnson, A database approach to course timetabling. J.Opl.Res.Soc. To appear
- 3. F. Glover (1986) Future paths for integer programming and links to artificial intelligence, *Comput.Oper.Res.* 13, 533-549
- 4. F. Glover & M. Laguna (1992) Tabu search. In Modern Heuristic Techniques for Combinatorial Problems, To appear
- 5. A. Hertz (1991) Tabu search for large scale timetabling problems, *Eur J.Opl.Res.* 54, 39-47
- 6. D. Abramson (1991) Constructing school timetables using simulated annealing: sequential and parallel algorithms, *Mgt.Sci.* 37, 98-113
- 7. D. G. Johnson (1990) Timetabling University Examinations. J.Opl.Res.Soc. 41, 39-47
- M. Lundy & A. Mees (1986) Convergence of an annealing algorithm. *Mathemat.Prog.* 34, 111-124



# Appendix E

This paper was presented at the ESRC Study Group on the Economics of Education, organised by Department of Economics and Pro-Vice Chancellor Brunel, The University of West London. The presentation took place on the 25th November, 1994 at Regents's College, London.

# The Construction of High School Timetables by Computer

# Zuhaimy H. Ismail and David G. Johnson Loughborough University Business School

### Abstract

Timetabling is a very hard problem and a general polynomial time deterministic algorithm is not known. The problem occurs in virtually every high school, college and university and the basic challenge is to produce a timetable for a given time period so as to avoid any class or teacher clashes while satisfying all the stated curriculum requirements and at the same time satisfying as many as possible of a range of additional 'medium' and 'soft' requirements. An expert system based on production rules may be useful for satisfying these requirements. An automatic computerised timetabling system with reasonable flexibility suited to the needs of most high schools is developed which provides a choice of feasible schedules and solutions in which the subjects allocated are well spread and satisfy the particular needs of an individual school. Collaboration with three schools in Leicestershire enables us to develop this system, which has been used to develop timetables for these schools.

## Introduction

Producing timetables is a regularly occurring problem in virtually all high schools, colleges and universities. The basic challenge is to produce a timetable over a given time period so as to avoid any violation of the hard constraints and at the same time satisfying as many as possible of the medium and soft constraints. The hard constraints are constraints that must be satisfied and a timetable will not be considered acceptable if they are violated, for example satisfying the specified curriculum requirements. The medium constraints may be violated if absolutely

necessary, but even then, this should be kept to an absolute minimum. An example here would be the need to teach particular subjects in double periods. The soft constraints are those where it is desirable if they can be accommodated within the timetable, but it is not considered critical if they are violated. For example the avoidance of certain subjects at key times during the day. Many of the routines have been developed to solve a specific problem at a particular school or establishment and relatively few systems have been used in more than one school without going through much local modification.

The system which has been developed is an interactive timetabling system for solving high school timetabling problems based on the concept of a hybrid between heuristic modelling and rule based expert systems. Rules are used to improve the assignment of subjects based on a range of soft requirements. A weight is associated with each rule so that the relative importance of each constraint may be adjusted by the user. The weight of the various requirements is a subjective factor effecting the system output globally and the solution generated by the system based on the specified weights might not satisfy fully the expectations of the user. To overcome some of these problems, the system is made highly interactive, allowing the user to compare and modify the output interactively on the screen.

### **The Problem**

The basic criterion in producing a timetable for high schools is to satisfy all the curriculum requirements and at the same time avoid clashes of subjects or teachers in any period. This problem has been well studied in the past and many approaches to the problem have been proposed. Early attempts were often based on integer programming models (Gotlieb, 1963, Lawrie, 1969), but were not particularly successful because there are usually too many variables and constraints. In recent years, alternative procedures have been developed and used to solve timetabling problems, often based on graph theoretical or more elaborate integer linear programming models (Tripathy 1984). Other authors have developed heuristic

procedures, some of them able to handle both hard and the medium constraints (Desroches, Hertz 1991, Kiaer and Yellen 1992.), but there have been relatively few examples of attempts to incorporate many of the soft constraints.

Timetabling problems differ widely, but they have several factors in common. First, most schools solve their timetabling problems by using human experts who have gained much insight into their own specific situation and are therefore able to create effective timetables based largely on experience and local knowledge. Furthermore, the constraints on the problem often change so frequently that any feasible solution is sufficient. Second, in solving these problems, quite often heuristics and rules of thumb are used which are sometimes very difficult to formalise mathematically. Third, even when an analytic formulation of the problem is attempted, many intricate constraints are not incorporated and only very simple constraints are used. In cases where a solution is not found, the expert (timetabler) often uses judgement in relaxing some of the constraints, something which is much more difficult within the framework of an analytical approach. Finally, timetabling problems are known to be impossible to solve by exhaustive search by machine (Abramson, 1991). These difficulties are present even in relatively small problems such as examination scheduling in a college.

## Constraints

The main consideration in generating a high school timetable is to satisfy all the curriculum requirements and at the same time avoid any class or teacher being 'double booked' in any period. A typical timetabling requirement for a high school taking pupils from 11-14 years in three separate year groups (years 7, 8 and 9) is that each year group follows a defined curriculum with typically 4 or 5 periods devoted to major subjects such as English or Science, down to just 1 period a week for minor subjects such as Music. The teaching week is divided into typically 25 to 40 periods (5 to 8 periods per day) and each period is devoted to one of 8 to 15 distinct subjects, with no free periods. For timetabling purposes, each year is

usually divided into two or three 'bands' with all the classes in each year band following an identical timetable and the school's staffing is such that in each department there are just enough teachers to teach each of the classes in one year band at the same time. Thus the timetable is basically planned on a 'band' basis rather than a 'class' basis and, for most subjects, all the pupils in a particular band will be taking the same subject in any period. These subjects vary according to the way the timetable is set up. In some schools, subjects like Humanities and English are combined to form one subject in timetabling. Staffing considerations therefore dictate that any subject can only be taught to at most one year band during any period. It would also be desirable if the number of occasions when any subject is taught more than once on the same day (apart from prescribed double periods) to any year band is minimised, and teacher free periods are spread as evenly as possible over the week.

As described earlier, however, there are also various soft requirements that are not critical in timetable construction, but it would be better if they could be satisfied. These soft requirements are not common to all high schools and invariably each school has its own special requirements. Therefore constructing a computer based timetabling system using a conventional programming approach would be very difficult and complicated. Five generic soft requirements are built into the system described here with the weights specified by the user. The system uses a hybrid of conventional programming and expert systems with the philosophy being to let the user decide which timetable features are required and the weight that is appropriate to each requirement.

## Procedure

Formulations of most real life timetabling situations as an integer programming problem or graph colouring problem have not been generally successful because of the large number of variables and constraints. Heuristic searches have had

reasonable success, but it is not easy to devise a heuristic which will be guaranteed to perform as well as an experienced timetabler. We therefore propose a combination of heuristics based on tabu search and simulated annealing with rule based expert systems, having the following properties.

- The algorithm is robust: it produces a feasible solution for a wide range of timetable variations such as different numbers of year bands, different numbers of daily periods, different numbers of subjects and different numbers of double periods.
- There is a variety of outputs to cover a range of timetable display formats.
- The solution is generated in a realistic time: it runs on a personal computer and users are able to execute several runs within a short time.
- The system is user friendly: it is easy to use with all instructions given on the screen.
- The timetable output can be displayed on the screen and modifications can be made interactively.

The process is divided into three main stages, namely to satisfy the hard constraints, the medium constrains and the soft constraints. The first stage is to randomly generate an initial timetable which satisfies the key constraint that no subject is repeated in the same period, and then systematically search for a feasible timetable that will satisfy all the curriculum requirements. This search is governed by an objective function that reflects the quality of the assignment and which aims to minimise repetition of subjects within a day. Some schools require certain subjects to be assigned as double periods and a special heuristic is developed to satisfy such requirements. Likewise, schools may require common periods to be included in the timetable where all classes are involved in the same activity simultaneously. The objective function is modified to force any double period or common period requirements in addition to minimising repeated subject assignments in any day. The search for an acceptable solution is based on the techniques of tabu search

(Glover 1989,1990) and simulated annealing (Van Laarhoven and Aarts 1987, Johnson 1990,1993).

Tabu search has its origins in combinatorial procedures applied to non-linear problems in the late 1970s and subsequently applied to problems such as travelling salesman, graph colouring, job shop sequencing and integrated circuit design (Glover, 1977). It is a metaheuristic procedure which seeks to find a global optimum to a combinatorial optimisation problem. It is an adaptive procedure with the ability to make use of many other heuristics, which it directs to overcome the limitations of local optimality. The core of the methodology is a short term memory function which prevents the return to a recently considered solution and thereby forces the solution away from a local optimum.

Simulated annealing originates from the analogy between the physical annealing process and the problem of finding (near) optimal solutions to discrete minimisation problems. The technique simulates the cooling of a collection of hot vibrating atoms. When the atoms are at a high temperature they are free to move around and tend to move with random displacements. However, as the mass cools the interparticle bonds force the atoms together. The process involves a series of random displacements from the current solution which are increasingly constrained to those which lead to an improvement in the objective being pursued.

The first two stages are completed when all the curriculum requirements, including the double periods and common periods, have been scheduled with subjects well distributed throughout the week. The user may wish to finish at this point and accept the feasible timetable which has been generated, or to proceed to the expert system phase. This is the phase where the soft requirements are identified and weights are allocated to each of the proposed rules. The rules cover most of the common soft requirements, the choice of which will vary from school to school. Table 1 gives the list of the rules currently available:

Rule 1	Two specified subjects should not be assigned next to each other
	throughout the week.
Rule 2	Two specified subjects should not be assigned on the same period
	throughout the week.
Rule 3	For a specified pair of year bands, two specified subjects should not be
	assigned on the same period throughout the week.
Rule 4	Two specified subjects should not be assigned on the same day
	throughout the week.
Rule 5	A specified subject should not be assigned to a specified sequence of
	periods on any day.

### Table 1: The rules which can be selected when constructing a timetable

Rule 1 is a requirement that prevents two particular subjects from being assigned consecutively. This rule is included because in some schools, for example, the timetabler may not wish to assign two active subjects consecutively, or two science subjects consecutively. Rule 2 requires two subjects not to be assigned on the same period throughout the week. This may arise if a teacher teaches more than one subject which is more likely to occur with minor subjects. In practice this type of requirement is often difficult to accommodate and will tend to severely constrain the timetable. Rule 3 is a special case of Rule 2. Instead of limiting the assignment for the whole timetable, this condition only applies to two specified year bands. For example, a particular teacher may be allocated to teach English to year band 1 and Drama to year band 2 which could lead to a clash if these two subject/year band combinations are not kept apart. Rule 4 is included to prevent two subjects . from being assigned on the same day. This rule allows the timetabler to distribute certain subjects throughout the week, for example to prevent PE and Games being assigned on the same day. Rule 5 is widely used in schools where a subject is prevented from being assigned to a period or a range of periods. This could be

used to prevent, for example, maths being taught in period 8 at the end of the day. Using conventional programming, it is difficult if not impossible to create an algorithm to cover all the potential requirements implicit in Table 1.

A heuristic is developed to satisfy each individual rule by simple swaps using tabu search. In most cases an individual rule can easily be satisfied. The problem arises when more than one rule is required. This is because the system will either override the previously assigned rule or, if overriding previous rule is prevented, it is often not possible to solve the problem due to the limited amount of movement available. To overcome this problem, the rules are paired and a heuristic is created to satisfy (as far as possible) each different pair of rules. If more than two rules are required, these must be assigned sequentially and to minimise overriding previously assigned rules, we develop a procedure based on the principle of least important first (LIF) or most important last (MIL). As described previously, each rule has a weighting factor between 0 and 3, with 3 being a highly desired rule and 0 as not required. If Rules 1,2 and 3 are selected with weights of 3,2 and 1 respectively, then rule 3 would be assigned first, followed by the heuristic combining rules 1 and 2. The procedure will continually evaluate all the rule violations and the procedure is repeated until all rules are satisfied or the pre-set stopping conditions have been reached.

### The System Implementation

The system is coded in Fortran and operates on a personal computer (PC). It is interactive, uses simple menu driven commands and has help options. One important feature is that the user has the ability to produce several alternative timetables quickly. In addition, the system includes an interactive feature that allows the user to modify the completed timetable to make minor modifications to an otherwise acceptable timetable.

The system generates three forms of output to assist the timetabler, namely

- A class timetable. This is a 'period by day' timetable for each year band indicating the lessons at each timeslot for each day
- A subject distribution timetable. This is a 'period by day' timetable for each subject indicating the class taught at each timeslot for each day.
- A daily timetable. This is a 'class by period' timetable for each day indicating the subject taught to each year band on that day.

## References

- Abramson D. (1991), 'Constructing School Timetable Using Simulated Annealing: Sequential and Parallel Algorithms', *Management Science*, Vol 37 (1), pp 98 - 113.
- Chahal N. and de Werra D. (1989), 'An Interractive System For Constructing Timetables', European Journal of Operational Research, Vol 40, pp 32 -37.
- Csima J. and Gotlieb C.C. (1964), 'Tests On A Computer Method For Constructing School Timetables', Communication of The ACM, Vol 7 (3), pp 160-163.
- de Werra D. (1985), 'An Introduction to Timetabling', European Journal of Operational Research, Vol.19, No.2, pp 151-162.
- Glover, F. (1987), 'Tabu Search', Technical Report, Center for Applied Artificial Intelligence, University of Colorado, Boulder, USA.
- Glover, F. (1989), 'New Approaches for Heuristic search: A Bilateral Linkage with Artificial Intelligence', European Journal of Operational Research, Vol. 39, pp 119-130.
- Gotlieb, C. (1963), 'The Construction of Class-Teacher Timetables', Proceedings of the IFIP Congress, Vol. 62, pp 73-77.
- Hertz A. (1992), 'Finding A Feasible Course Schedule Using Tabu Search', Discrete Applied Mathematics, Vol 35, pp 255-270.

- Hertz A.(1991), 'Tabu Search For Large Scale Timetabling', European Journal of Operational Research, Vol 54, pp 39-47.
- Hertz, A. and de Werra, D. (1987), 'Using Tabu Search Techniques for Graph Coloring', Computing, Vol 39, pp 345-351.
- Johnson, D. (1990), Timetabling University Examinations', Journal of the Operational Research Society, Vol: 41 (1), pp 39-47.
- Johnson, D. (1993),'A database approach to course timetabling', Journal of the Operational Research Society, Vol: 44 (5), pp: 425-433.
- Laporte G. and Desroches, S. (1986), 'The Problem of Assigning Students to Course Sections in Large Engineering School', Computer and Operations Research, , Vol 13, pp387-394.
- Lawrie, N.L & Veitch H. (1975), 'Timetabling and Organization in Secondary Schools', CEER Publishing Company, New Jersey.
- Lawrie, N.L (1969) An Integer Programming Model of a School Timetabling Problem, Computer Journal., Vol 12, pp 23-46.
- Lewis, C.F. (1961), 'The School Timetable', Cambridge University Press, Cambridge
- Tripathy A. (1984), 'School Timetabling A Case in Large Binary Integer Linear Programming', *Management Science*, Vol 30(12), pp 1473-1489.
- Tripathy A.. (1992), 'Computerised decision Aid for Timetabling A case Analysis', Discrete Applied Mathematics, Vol 35, pp 313-323.