CHAPTER I
INTRODUCTION

1.1 Introduction

The acronym EMIS stands for ‘Education Management Information System’. It is a system designed to systematically organize information related to the management of educational development. In the words of K.C. Tung, ‘EMIS refers to a System for Processing Information for the Management of education resources and services’.

EMIS is responsible for the promotion and use of information for policy planning and implementation, decision–making, and the monitoring and evaluation of an education system. As we live in the age of information, the success in organizing information systems for the development of education lies in the use of information for development. Not using accurate and timely educational information for monitoring development activities results in retarded development.

EMIS also substantially aids efforts made to assess the performance of an education system. It also closely monitors the equitable distribution of resources, and plays an active role in providing information to top management about the deployment of teachers, student performance assessment, resource allocation and internal efficiency of the education system.

In Technical and Higher Education no efforts have been made to strengthen Management Information system in the recent past. The available information on higher education is limited and outdated. More specifically up-to-date information is not available on many desired variables. If made available it can be of great help in better planning and management of education. As of now, it seems that there is no regular mechanism to develop plans specifically for Higher Education at any level. The disaggregated data when available will encourage all providers to plan and manage affairs of Higher Education better.
Various managerial activities in an educational institution especially in higher education institution like college and university are given below.

1. Campus Management
2. Department Management
3. Admission Management
4. Student Management
5. Staff Management
6. Library Management
7. Accounts Management
8. Inventory Management
9. Hostel Management
10. Transport Management
11. Examination Management
12. Laboratory Management
13. General Management

Department Management activities of an institution include Department History, Course Management, Syllabus Management and the Course timetable. In this thesis various efficient techniques for the creation of the Institution’s Course timetable are proposed.

1.2 Timetabling Problem

A timetabling problem is a kind of problem in which events (classes) have to be allotted into a number of timeslots, subject to various constraints. The need for powerful methods for solving a large timetabling problem is plain by considering the simple fact that with, say, e classes to be fitted into t timeslots, there are $t^e$ possible candidate timetables, which vary in optimality according to the constraints of the problem.

Timetabling problems are particularly challenging and can be solved using Artificial Intelligence and Operations Research techniques. They are
problems of time-based planning and combinatorial optimization which tend to be solved with a cooperation of search and heuristics, which usually lead to satisfactory but sub-optimal solutions. Why are timetabling problems so difficult?

1. They are computationally NP-complete problems. This means that there is no known deterministic polynomial time algorithm. Also, the space of possible solutions for most real problems is far too large for brute-force or undirected search methods to be feasibly applied.

2. Advanced search techniques using various heuristics to prune the search space will not be guaranteed to an optimal (or near optimal) solution. In other words, it is very difficult to design effective heuristics.

3. Timetabling problems are often complicated by the details of a particular timetabling task. A general algorithmic approach to a problem may turn out to be inapplicable, because of certain special constraints required in a particular instance of that problem.

4. Real world timetabling problems often involve constraints that cannot be precisely represented or even precisely stated.

Conventional computer-based timetabling methods concern themselves more with simply finding the shortest timetable that satisfies all the constraints, usually using a graph-coloring algorithm and less with optimizing over a collection of soft constraints. That is to find sets of classes that can be scheduled at the same time correspond to finding a coloring such that adjacent nodes have different colors: each color represents a timeslot, and each edge a constraint that the two vertices which it connects must occupy different slots (have different colors). Knowledge-based and OR-based approaches to solving such problems are hard to develop, are often slow and can be inflexible because the architecture itself was based on specific assumptions about the nature of the problem. There are commercial software tools available, described in (Tim Duncan, 93), which can be used to build timetabling applications. It is more
difficult to handle soft constraints such as preferences in these tools because they are usually based on straightforward constraint satisfaction techniques. Most often people still resort to hand-crafted solutions starting from an earlier solution and making a sequence of modifications to cater for changed requirements. This typically leads to sub-optimal solutions that bring significant organizational or financial penalties.

1.3 Genetic Algorithms (GAs)

Genetic Algorithms (GAs) are a group of methods which solve problems using algorithms inspired by the processes of neo-Darwinian evolutionary theory. In a GA, the performance of a set of candidate solutions to a problem (called 'chromosomes') are evaluated and ordered, and then new candidate solutions are produced by selecting candidates as 'parents' and applying mutation or crossover operators which combine bits of two parents to produce one or more children. The new set of candidates is then evaluated, and this cycle continues until an adequate solution is found. In chapter 2 and chapter 3 GAs are introduced and explained in more detail.

1.4 Constraint Satisfaction Problems

Constraint Satisfaction Problems (CSPs) involve a set of problem variables, a domain of potential values for each variable and a set of constraints specifying which combinations of values are acceptable (Lhomme, 1993). A solution specifies an assignment of a value to each variable that does not violate any of the constraints (Mackworth, 1994; Freuder and Wallace, 1994). A large number of problems in Artificial Intelligence (AI) and other areas of computer science can be viewed as special cases of CSP. Some examples are machine vision, belief maintenance, scheduling, temporal reasoning, graph problems, floor plan, the planning of genetic experiments, circuit design, machine design and manufacturing, and diagnostic reasoning (Kumar, 1992).
1.5 Motivation

Timetabling problems, particularly course timetabling is a difficult task faced by educational institutions. Solving a real world timetabling problem manually often requires a large amount of time and expensive resources. In order to handle the complexity of the problems and to provide automated support for human timetables, much research in this area has been invested over the years.

A brief observation of the recent timetabling literature (see Chapter 2) shows that, a wide variety of papers, from the fields of operational research and artificial intelligence, have addressed the broad spectrum of timetabling problems. Early timetabling research focused on sequential heuristics which represented a simpler and easier method for solving graph colouring problems, the principle idea being to schedule events one by one starting with the most difficult first. In recent years, interest in meta-heuristic approaches such as simulated annealing, tabu search and genetic algorithms has increased due to the ability of these approaches to generate solutions which are better than those generated from sequential heuristics alone. Normally in timetabling, an initial solution is constructed using an appropriate heuristic, and then the improvement is carried out using these meta-heuristics. However, some meta-heuristics are dependent on certain parameters. For example, simulated annealing depends on a cooling schedule; tabu search requires (among other parameters) an appropriate length of tabu list. The performance of these approaches may vary from one instance to another which might depend on the setup of these parameters, the neighborhood structure and the search algorithm itself. Many meta-heuristic methods work well on certain problem instances but often are not readily applicable and are expensive to adapt to new problems. Hence, the development of a more general framework that can work effectively across different problems has been studied in recent years.
To this end, the area of hyper-heuristics (i.e. heuristics to choose heuristics) is currently being investigated in contrast to some meta-heuristic research which focuses upon comprehensive problem-specific knowledge to arrive at good solutions for specific problems. In the last few years, researchers have also attempted to investigate knowledge-based such as case-based reasoning, and constraint-based reasoning for solving timetabling problems. This thesis is derived from an interest in developing automated algorithms to tackle timetabling problems in a more effective way than currently exists. In this research work, as a different approach, in Genetic Algorithms, Constraint Satisfaction algorithm is applied to handle constraints. In this thesis, more general algorithms that can work effectively across different educational institutions course timetabling that will reduce computational effort and give alternate solutions are to be attempted.

1.6 Conclusion

The objective of this thesis is to establish an efficient method using genetic and constraint satisfaction algorithms for the timetabling problem. Having discussed the complexity of the timetabling problem, in the following chapter the various existing methods to solve the timetabling problem are explained.