**A TABU SEARCH ALGORITHM TO SOLVE A COURSE TIMETABLING PROBLEM**

**Projectees:**

**Dashrath N. Gavhane**

**Amit N. Maske**

**Pratik N. Sadawarte**

**Pooja A. Gumphawar**

**Shweta R. Ramteke**

**Abstract**

University course timetabling problems must be solved by the administration every year, or even term, and they involve a large amount of human and material resources. In the literature, the problem formulation does not contain the constraint that there should be no conflict between lessons in the same section. In this paper it is shown how a course timetabling problem which also includes this constraint can be formulated, and a tabu search algorithm is proposed to solve this problem. To show the effectiveness of the proposed algorithm, it is applied to the timetabling problem of the Statistics Department of Science University using a computer program based on this algorithm.

**1. Introduction**

The problem of building a university timetable consists of assigning instructor-course- room combinations into specific time periods. The objective in a classical course timetabling problem is to reduce the number of conflicts, which occur when courses involve common students, common teachers or require the same classrooms. For large institution such as universities, the problem becomes more difficult since additional constraints have to be taken into account.

 Solving a course timetabling problem is very difficult. The main difficulty is related to the size of the problem. It involves a large number of students, teachers, courses and rooms, linked in many ways by objectives and conditions, and therefore each solution procedure must take into account very large number of variables and constraints. Moreover, the structure of the timetables varies from university to university, due to differences in the education systems. Even within a university, there are major differences among departments depending on the particular ways in which teaching is organized. For these reasons, during the last few decades many contributions related to course timetabling have appeared and a huge variety of timetabling models have been described in operations research literature.

**2. Literature Review**

 The Linear and Integer Programming techniques, the first applied to timetabling, were developed from the broader area of mathematics programming. Mathematical programming is applicable to the class of problems characterized by a large number of variables that intersects within the boundaries imposed by a set of restraining conditions (Thomson,1967)[1]. The word “programming” mean playing in this context and is related to the type of application (Feiring, 1986)[2]. This scheme of programming was developed during World War II in connection with finding optimal strategies for conducting the war effort and used afterward in the field of industry, commerce and government services(Bunday,1984)[3].

 Linear programming is that subset of mathematical programming concern with the efficient allocation of limited resources to know activities with the objective of meeting a desired goal such as maximizing project or minimizing costs(Feiring,1986)[2]. Integer programming deal with solution of mathematical programming problem in which some or all of the variables can assume non-negative integer value only. Although LP method are very valuable in formulating and solving the problem related to the efficient use of limited resources they are not restricted to only these problems (Bunday,1984)[3]. Linear programming problem are generally acknowledged to be efficiently solved by just three methods, namely the graphical method, the simplex method, and the transportation method (see eg., Palmers and Innes 1976; Makower and Williamson, 1985)[4].

**3. Time table Implementation**

 This application has been developed using Java as front end tool and MySQL Server as its back end tool. The application has been coded to be platform independent running on Java Virtual Machine.

 Netbeans IDE has been chosen as its development environment because of the following features

* Designing interface for the application has been simplified by its drag and drop GUI pallet.
* Debugging can be easily done using the Logger class.
* Easy database access with NetBeans database plugin.
* Simplified automated editor error detection.
* Automatic code generation.
* Automatic documentation.
* Simplified class factory method lookups.
* Easy to create jar files using build option.
* Profile option.
* Project can be run on debugging mode which provides current state of the variables with the help of break points.
* NetBeans IDE has wide help and support on the web.

While coding TimeGene application several constraints related to its computation has been taken into account.

Timetable generating problem provides us with various alternatives in the design of the algorithm, interface and the database. Among the various designs what we have implemented is detailed below:

**Interface Implementation:**

There are ten classes each contains a JFrame which is associated with an interface.

The association are as follows

* ClassroomScheduler.class for login interface
* Classroom.class for basic information interface
* TimeSlot.class for schedule interface
* Courses.class for subject interface
* Batches.class for batches interface
* ModifySchedule.class for Timetable Output interface
* Conflict.class for open and save interface

**4. Algorithm Implementation**

TimeTable generation is an NP-Complete problem; specifically speaking NP-Hard. So it lacks a proper time bound for execution i.e. problems like these often can have many different outputs. So we assign cost to each output which gives the measure of deviation of the output from the desired one. So our aim is to get the output with minimum cost if there is one.

Genetic algorithm can give best results but the time needed for it to compute cannot be determined so we have developed an alternative approach which can be applied to solve most of the NP-Complete problems.

* First determine the various constraints which the output must satisfy.
* We then categorize them as soft and hard constraints.
* Third step is to make a procedure which can generate an output for most of

the possible inputs.

* The final step is to reduce the cost.



The current working scenarios these can be explained as follows

The first two steps are explained earlier. The procedure mentioned in the third step here is the gene() fuction in the selbat class. What this function does is to assign priorities to teacher, subject and position. So that if we arrange timetable according

to this priority there is a greater probability to end up in the output which satisfies all the hard constraints. First and foremost priority is that of teachers. Subject priority and position priority depends on teacher priority, also teacher is the most important resource in the time table.

**5. Experimental Results:**

**Fig 1: List of Professor**

****

**Fig 2: Time Slot**



**Fig 3: Courses**

****

**Fig 4: Modify Schedule**

****

**Fig 5: View Chart**



**Fig 6: Conflicts**



**6. Conclusion**

 Tabu search algorithm produces very good time tables which do not contain conflicts.

 Therefore, since the proposed tabu search algorithms effective in solving the present problem, it can be used to solve timetabling problem of other Departments or any timetabling problem in Universities.

**7.References**

* Alvarez-Valdes, R. et al. Assigning students to course sections using tabu search. *Annals of Operations Research*. Vol. 96 (2000) p. 1-16
* Bozkaya, Burcin. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*. Vol. 144 (2003) p. 12-26**.**