UCS 2201 Fundamentals and Practice of Software Development

TIME TABLE MANAGEMENT SYSTEM FOR AN ACADEMIC INSITUTION

ABSTARCT:

The timetable is necessary tool for the efficient working of a university. It is really a mirror that reflects the entire educational programme of the university. Timetable concern all activities with regards to producing a schedule that must be subjective to different constraints' college timetable is a temporal arrangement of a set of lectures and classrooms in which all the given constraints are satisfied.

This paper introduces a framework for a highly constrained school timetabling problem, which was modelled from the requirements of various Finnish school levels. We present a success-full algorithm to solve real-world problems as well as artificial test problems. Moreover, we find the best configuration for this algorithm using brute force and statistical analyses. Finally, we pro-pose a set of benchmark problems that we hope the researchers of the timetabling problems would adopt.

Class timetable: Timetable generation usually helps in preventing the wastage of time for the student in the search of their classes and teachers. It shows the distribution of subjects in each class along with teachers for each period. it also indicates the breaks in between the teaching period along with recess.

Teacher's timetable: Timetable generation for teachers is also much important in which it ensures equitable distribution of work among teachers.

The timetable is needed to be scheduled in such a way that the number of different courses with a number of subjects in each, handled by a limited faculty provided with their slots and timings does not overlap.

INTRODUCTION:

Timetable generation which generates timetable or schedules for students or faculty which reduces the manual work.

Once the inputs like faculty, subjects, no of classes, labs etc ate given it will generate the period slots for the entire week and also the substitutional hours.

Previously timetable was manually made by the head of the department involves a lot of manual work and is a time consumptive process.

The system will help the colleges to generate the timetable automatically without any manual workload and saves a lot of time that plays a prominent role in the present lifestyle. While scheduling even the smallest constraints can take a lot of time and the cases are even worse when the number of constraints or the amount of data to deal with increases.

EXISTING WORK:

GRAPH COLOURING APPROACH

This algorithm colours the graph such that no two adjacent vertices are of same colour. The slots are assumed to be the colours. After first step the results obtained are the courses and the slots in which the courses are to be scheduled. For solving Course Timetable scheduling problems using graph colouring, the problem is first formulated in the form of a graph courses act as vertices.

Depending on type of graph created, edges are drawn accordingly. One type is conflicting graph where edges are drawn between conflicting courses having common students. Other one is non-conflicting graph, where edges are drawn between mutually exclusive courses having no students in common.

This two-step method is efficient in certain cases, while in some conflict graph is created direct. In such cases, the conflict graph must consider course, teacher and room conflicts simultaneously.

As mentioned earlier, there can be various aspects of a scheduling problem. When teachers are involved in resources, other factors like availability of teachers, subject area preferred by each teacher acts as additional data inputs which needs to be provided for making a complete schedule.

HEURISTICS APPROACH

Heuristic optimization methods are explicitly aimed at good feasible solutions that may not be optimal where complexity of problem or limited time available does not allow exact solution. The empirical evaluation of heuristic method is based on analytical difficulty involved in the problem's worst -case result.

In its simplest form the scheduling task consists of mapping class, teacher and room combinations onto time slots. One possible approach is as follows: We define a tuple as a particular combination of identifiers such as class, teacher and room, which is supplied as an input to the problem.

For each period of the week, we make a count of the number of occurrences of each class, teacher and room identifier. The cost of the entire timetable is the sum of each of the individual cost It takes the user input of a number of subjects, number of teachers, subjects every teacher takes, number of days in a week for which the timetable needs to be set, number of time slots in a day and the maximum lectures a teacher can conduct in a week.

GENETIC ALGORITHM METHOD

We are going to use Genetic Algorithm to solve class scheduling problem with custom selection methods. Genetic algorithm is population based heuristic method largely applied to solve the scheduling problems. An individual is characterized by a set of parameters (variables) known as genes. Genes are joined into a string to form a chromosome (solution). We model our solutions as chromosomes.

In our example scenario, chromosomes will be different lecture systems for students taking different modules. Consider that we have to coordinate student groups, modules, lecture halls, the days of the week and time. You can represent a lecture session as you can encode the lecture session as a binary pattern to a chromosome.

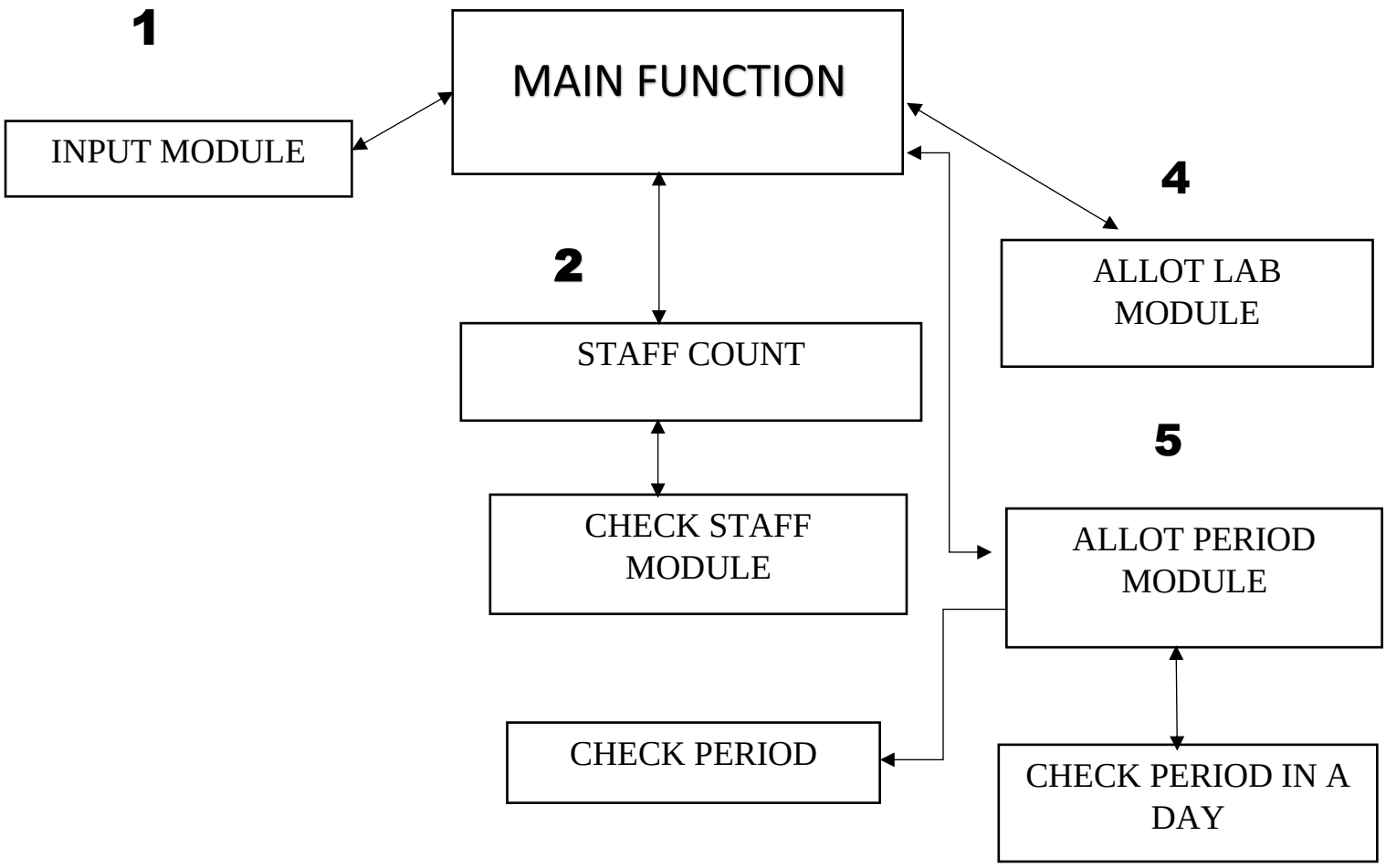
BRUTE FORCE METHOD:

Brute Force Algorithms are exactly what they sound like – **straightforward methods of solving a problem that rely on sheer**

computing power and trying every possibility rather than advanced techniques to improve efficiency.

Brute force algorithm is a technique that guarantees solutions for problems of any domain helps in solving the simpler problems and also provides a solution that can serve as a benchmark for evaluating other design techniques, but takes a lot of run time and inefficient.

ARCHITECTURAL DESIGN:



MODULE DESIGN:

1.INPUT MODULE:

Gets all the required inputs like staff, name of the subject, semester, number of classes and labs using the structure.

2.STAFF COUNT MODULE:

Initially counts the total number of staff with repetition and calls check staff module.

CHECK STAFF MODULE:

Count the number of staff without repetition using string function and store the name of the staff in a new structure “staff_tt”.

3.LAB COUNT MODULE:

Counts the total number of staff with repetition and calls the check lab module.

CHECK LAB:

Count the number of staff without repetition using string function and store the name of the staff in a structure “lab_tt”.

4.ALLOT LAB MODULE:

1.Generate a random number and allots a lab if it is free and if the constraints are satisfied.

2.Store the allotted lab class in a allotted period in the respective staff timetable.

5.ALLOT PERIOD MODULE:

1.Generate a random number and allot a period if the period is free, then calls check period and check period in a day module and verifies the constraints.

2.Stores the name of subject in an allotted period for a respective staff.

CHECK PERIOD:

1.Check the previous timetable and allots the class without clash.

CHECK PERIOD IN A DAY:

Checks for a repetition of period in a day.

IMPLEMENTATION:

OUTPUT:

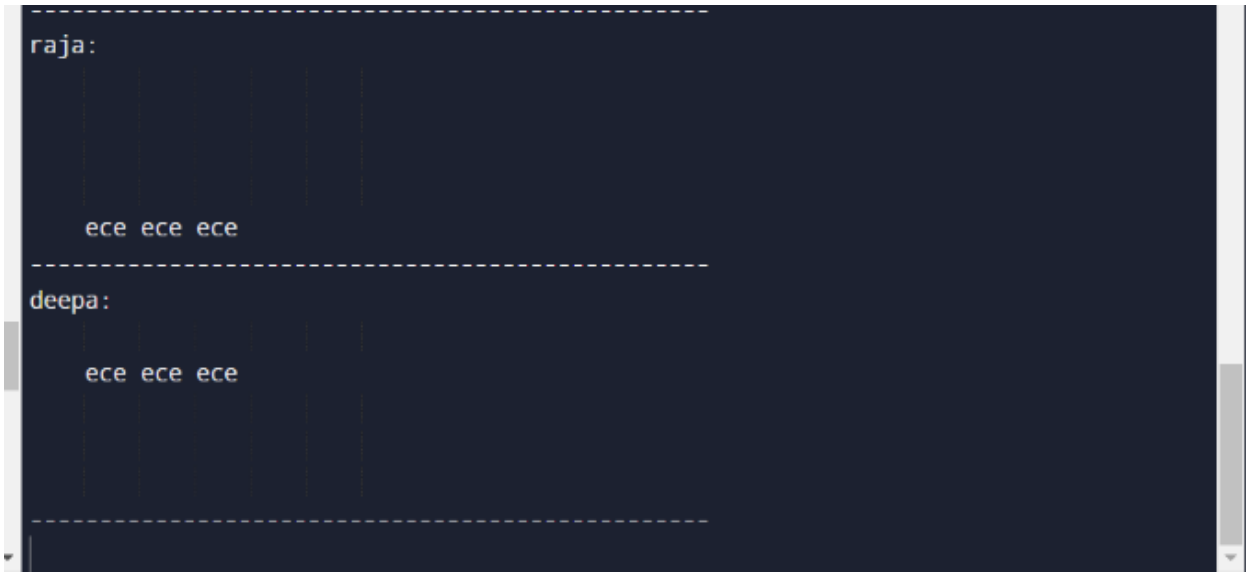
```
/tmp/UfdGGLoUS0.o
ENTER NUMBER OF CLASS:2
------------------------------Input for class 1------------------------------
Enter semester:5
Enter Class:cse
ENTER THE NUMBER OF THEORY CLASS:   4
ENTER NUMBER OF LAB CLASS:  2
Enter theory subject 1:evs
Enter faculty for evs theory:priya
Enter theory subject 2:math
Enter faculty for math theory:yugesh
Enter theory subject 3:fpsd
Enter faculty for fpsd theory:kanchana
Enter theory subject 4:data
Enter faculty for data theory:senthil
Enter lab subject 1 :clab
Enter faculty for clab lab:kanchana
Enter lab subject 2 :mech
Enter faculty for mech lab:babu
*****Enter number of periods per weeks*****
Enter number of period for evs:3
Enter number of period for math:5
Enter number of period for fpsd:4
Enter number of period for data:3
Enter number of period for clab lab:1
Enter number of period for mech lab:1
```

```
------------------------------------------------------------------
----------------------------Input for class 2----------------------------
Enter semester:7
Enter Class:ece
ENTER THE NUMBER OF THEORY CLASS:   3
ENTER NUMBER OF LAB CLASS:  3
Enter theory subject 1:java
Enter faculty for java theory:angel
Enter theory subject 2:math
Enter faculty for math theory:yugesh
Enter theory subject 3:evs
Enter faculty for evs theory:murugesh
Enter lab subject 1 :mech
Enter faculty for mech lab:kala
Enter lab subject 2 :civil
Enter faculty for civil lab:raja
Enter lab subject 3 :elec
Enter faculty for elec lab:deepa
*****Enter number of periods per weeks*****
Enter number of period for java:3
Enter number of period for math:4
Enter number of period for evs:2
Enter number of period for mech lab:2
Enter number of period for civil lab:1
Enter number of period for elec lab:1
------------------------------------------------------------------
math    fpsd    BREAK   mech    mech    mech    data
```

```
------------------------------------------------------------------
math    fpsd    BREAK   mech    mech    mech    data
evs     fpsd    data    BREAK   math
evs math    BREAK   clab    clab    clab
evs     data        BREAK   fpsd    math
math    fpsd            BREAK
--------------------------------------------------
java    evs BREAK   mech    mech    mech    math
math    elec    elec    elec    BREAK
math                BREAK
java    mech    mech    mech    BREAK   evs
java    civil   civil   civil   BREAK       math
--------------------------------------------------
priya:

cse
cse
cse

--------------------------------------------------
```

```
-------------------------------------------------
yugesh:
cse                     ece
ece                 cse
ece cse
                        cse
cse                     ece
-------------------------------------------------
kanchana:
    cse
        cse
            cse cse cse
                    cse
    cse
-------------------------------------------------
senthil:
                        cse
            cse

        cse
```

```
-------------------------------------------------
babu:
            cse cse cse




-------------------------------------------------
angel:
ece
ece
ece
-------------------------------------------------
murugesh:
    ece


                    ece

-------------------------------------------------
kala:
            ece ece ece


    ece ece ece

-------------------------------------------------
```



CONCLUSION:

The major benefit of this project is to store the given inputs and generate a timetable following the given constraints. Instead of tedious paper work, students, staff and common people can view the timetable with a quick turnaround. This system is user friendly and provides faster and better generation of timetable, which in turn saves time. There are few points that justify the need of this system:

- user friendly
- faster and better generation of timetable
- Saving time and manpower

REFERENCE:

[1] E.K. Burke and P. De Casemakers (2003). The Practice and Theory of Automated Time-

tabling IV: Revised Selected Papers from the 4th International conference, Gent 2002,
Springer Lecture Notes in Computer Science, vol. 2740, Springer.

[2] E.K. Burke and M. Trick (2005). The Practice and Theory of Automated Timetabling V: Re-

vised Selected Papers from the 5th International conference, Pittsburgh 2004, Springer Lecture
Notes in Computer Science, vol. 3616, Springer.

[3] E.K. Burke and Hana Rudová (2006). Proceedings of the 6th International Conference on the
Practice and Theory of Automated Timetabling, Masaryk University.

BY TEAM HASH INCLUDE

1)THIRISHA M- 3122 21 5001 056

2)L SRI SHALINI- 3122 21 5001 047

3) TEJASWI KAKARLA- 3122 21 5001 039