11th IEEE UAE STUDENT DAY

Saturday May 21st, 2016

Software Engineering Project (SEP) Competition

Intelligent Time Table Schedule: ITTS

Contents

I. Competition Rules

1. Only **IEEE UAE Students Members** are eligible for this competition.
2. Each institution can submit a maximum of two entries in this competition.
3. Each competing team shall have no more than three students.
4. The contestants shall not use any unauthorized or unlicensed software.
5. The source code and all relevant documentation shall be made available to the competition coordinator and the judges on the day of the competition.
6. Salient features of the software shall be documented with the aid of an A1 size Poster. A brief user manual shall also be provided.

II. Project Summary

The rapid technological advances in hardware and software have led to the emergence of the computing paradigm, where information is accessible anywhere and at any time. One of these paradigms is the Intelligent Time Table Schedule (ITTS) which constitutes an important module for educational institutions. In this module, the distribution of the load has to be fair for lecturer in terms of experience and time management; and have to take in consideration the number of students, available resources, constraints ... etc. In addition, the module could help students to take a chance to enroll in suitable courses according to their levels. This fairness will be applied by proposing an ITTS technique to produce such table, where users preferences including students and instructors, will be used and optimized for this purpose.

III. General Information

Timetabling is a really hard problem that belongs to a larger-area of scheduling. Scheduling is usually described as the process of resources allocation over time to accomplish a list of activities. Course timetables scheduling in a college/university is a common task. Generally, this task is performed manually and takes several days (sometimes weeks) of iterative improvements based on the staff and student feedbacks.

The objective of this project is to implement a software that solves the scheduling challenges of an educational institution and generates a timetable by using intelligent algorithm. The algorithm should take into consideration students and staff data for a department (e.g. courses, lecturers, days, classrooms, lecture duration, and some given constraints). The software should schedule the instructor and the students for each suggested course within a specified time periods in a particular classroom, while satisfying as many of the required constraints as possible.

In summary, the system is supposed to have or do the following:

- Students should be able to enter their preferences of courses.

- Instructors should be able to enter their preferences of offered courses.
- The system should have a list of constraints that are respected as much as possible during the generation of suggested time tables. The constraints could include but are not limited to:
  - ✓ Prerequisite courses and study plan according the major of students.
  - ✓ Students limit of courses per semester and instructor load limit too.
  - ✓ Availability of physical resources such as a classroom. Also the capacity of the classroom should be considered.
- Any other relevant suggestion to the proposed system as described will be considered.

IV. Technical Specifications

The timetable solution is a combinatorial optimization solution that consists of some constraints which should be considered as explained in the following table:

Preferences	Constraint
Classroom	• The maximum number of students per classroom. (should vary from one classroom to another)
Instructor	• The number of lectures should not exceed the instructor teaching load (4-5 Courses per instructor
Student	• The total number of enrolled courses should not exceed 18 CR.H. (between 3-6 courses per student)

V. Functionality

- Does the application run properly?
- Does the software generate acceptable timetables?
- Does the software solve the constraints conflict?
- Does the software respect all constraints?
- How long it takes for the software to generate a time table?
- How practical is the timetable that is displayed by the software? For example, is the suggested timetable is closer to that created manually by a college/university?

VI. Theoretical Knowledge

- Can the student explain timetable scheduling problem?
- Can the students show sufficient understanding of intelligent algorithm concepts?
- Can the students show sufficient understanding of the methodology and designing process they used for implementing their selected intelligent algorithm?
- Did the students identify and understand advantages and disadvantages of their developed software?
- Can the students demonstrate understanding of how they can extend the system functionalities and overcome the limitations in future work?

VII. User-Friendly Interface

- What is the users' perception of the developed software? (clear interface, menu options, and appropriate visual elements)
- Is the software easy to use by users with no computing background?
- Is the system user friendly for entering data, and how fast and practical is it to enter data?
- How easy is it for students to enter their preferences of courses, and how easy is it for instructors to insert preferences and constraints?

VIII. Poster and User Manual

The students are required to provide:

- A brief user manual.
- An A1 size poster to provide a concise software description, and depict the technical specifications / methodology / tools / techniques used in the development of the software.

IX. Testing Procedure

Student teams will be asked to demonstrate their software. The judges, will review the software data including the constraints that are used for demonstration and testing. Each team will demonstrate their software using a sample input of their own according to the specifications shown in the table below. The judges will test the software in a way that is uniform for all teams. The judges will also ask questions concerning the theoretical knowledge related to the developed software and its user interface. They will also evaluate the poster and the brief user manual. Each team will be then given 15 minutes to explain and demonstrate their software according to the given specifications. Additional features, if they exist, will be evaluated too. Following are a list of

minimum number of tables, minimum number of records per table, and a list of minimum required fields for each table.

	No.	Table	No Records	Fields
Input	**1**	Instructors	8	Instructor-ID, Name(s), Major, Load
	2	Students	110	Student-ID, Name(s), Major, Load
	3	Study-Plan-Courses-&-Prerequisite-&-Instructor	41	Course-ID, Title, Credit-Hrs, Prerequisite, Taught-by-Instructor
	4	Lecture-Rooms	7	Hall-ID, Capacity
	5	Time-Slots	7 per day , 1.5 Hrs. each, (4 days): Sun & Tue, Mon &Wed	Slot-ID, Days, Period
	6	Student-Transcripts	Varies between 5-35 Courses for each Student	Student-ID, Course-ID, Term, Year, Grade
	7	Preferred-Students-Courses-for-Term	5-6 Course-per-Student	Course-ID, Student-ID
	8	Instructor-Suggested-Courses-for-Term	1-4 Courses-per-Instructor	Course-ID, Teacher-ID
Output	**1**	Time-Table	As Needed	Course-ID, Course-Title, Instructor, Classroom, Time, Student-ID, Student-Name

Please note the following:

- When generating a dummy data for the project's tables, try to vary your values in a way that is analogous to real-life values. For example, the Transcript table could have 1/4 first year students, 1/4 second year students, 1/4 third year students, and 1/4 forth year students. Not evenly, it could be a little more or less.
- For Simplicity, each course is assigned to only one instructor.
- You have to suggest on Number of Offered Courses and Number of Students per Class ... etc. They could be chosen randomly or manually.
- Number of sessions per course should vary between 1 and 2. Of course, most courses would have only 1 session.
- The output table may be split into two or more tables to avoid redundant data. For example, the first table may include only number of students in a class along with the class information.

X. Evaluation

A panel of three judges, to be selected by the IEEE UAE Students Day Steering Committee, will assess the entries of the competition. The competition criteria that will be used for judging the entries are given below:

No.	Evaluation Criteria	Mark	1st Judge	2nd Judge	3rd Judge	Average = Judges (1+2+3) / 3
1	**Functionality**	50				
2	**Theoretical Knowledge**	15				
3	**User-friendly Interface**	15				
4	**Additional Features**	10				
5	**Poster and User Manual**	10				
	Total Mark	**100**				