A Dynamic System for Real-World University Examination Timetabling Problem Using AAC approach Case Study: Taibah University-CCSE

Anmar F. Abuhamdah
Department of Management Information Systems, College of Business Administration, Taibah University, Almadinah Almunawarah, Kingdom of Saudi Arabia.

Abstract

Generally, scheduling consider as NP hard problem in almost universities and education. There are different types of scheduling problems concerning with assigning set of events to specific rooms and timeslots subject to variety of constrains. Different education scheduling like exam scheduling needs different systems to deal with different constrains. Each year, every individual college has to design its own exam timetable. Exam timetabling dealt with a set of courses, rooms, and timeslots while satisfying some predefined constrains. Constrains classified as hard that must be satisfied to obtain feasible timetable or soft that may satisfy to improve the timetable while ensuring the feasibility. Therefore, in this paper, a dynamic real system is proposed that may use in different college or university. In this paper, a graph coloring used to obtain feasible timetable and adaptive acceptance criteria used to improve the timetable in the improvement phase. In order to evaluate the proposed system, previous data for CCSE in Taibah University is used as the test domain, to ensure that the proposed system timetable is comparable to the previous timetable advertise by the college. Outcome shows, that the system generate good quality timetable.

Keywords: Scheduling, Exam Timetable; Construction phase; Adaptive acceptance criteria; Taibah University.

INTRODUCTION

Scheduling (timetabling) problems are considered as a type of the optimization problems, in which assigning some events to particular time and venues subject to variety of constrains [1-4]. Timetabling problems are highly constrained and difficult to be solve for optimality by using a specific technique, which considered as a challenge for artificial Intelligence and Operations Research techniques, meanwhile, its considered as NP (nondeterministic polynomial time) hard problems [1-3]. Constrains in timetabling problems are classified as a hard and soft constrains [5], where the hard constrains must be satisfy to obtain feasible solution for the problem and the soft constrains desired to be satisfy to minimize the timetable to obtain better solution quality. There are several type of timetabling problems such as; educational timetabling (e.g. university course/exam timetabling), transportation (e.g. bus/train timetabling), industry (e.g. surgeon timetabling), and sports (e.g. pairs team matches timetabling) [4-6].

Educational timetabling involves of allocating a set of events (e.g. courses, exams, etc) into a fixed number of timeslots and venues, while minimizing the violation of a set of constrains. Educational timetabling classified into school or university timetabling. In school timetabling, every class has its own students with a fixed number of students, where in university timetabling different class may have common students and each class have different number of students, so we have to ensure the size of room, which is more difficult. University timetabling problems classified into course or exam timetabling problems, however, these problems consider similar to each other and it is difficult to differentiate between them except in the problem structure [6]. Exam timetabling problems deals with scheduling a number of sets [7-10]. These sets are divided into: courses exams timetabling, students scheduling, classroom assignment and timeslots.

Generally, the implementation phase in the methodology consists of two phases for solving scheduling or timetabling problem [11-12], first, satisfy the hard constrains in the construction phase to obtain feasible solution and secondly try to minimize the soft constrains in the improvement phase to obtain better quality solution.

In the construction phase, a sequential graph coloring algorithm [8, 10] is used in order to construct feasible solution (initial solution). However, there are various graph coloring algorithm (i.e. Largest Degree first, Largest Color degree first, …etc) can be used in the construction phase. Wherein the improvement phase, an approach or algorithm is use on the initial solution to minimize the violation of soft constrains while ensuring the feasibility.

Many researchers classified approaches into several categories [10, 13-16]; however, these approaches investigated over different optimization problems. In recent years, there are variety of approaches have been applied over timetabling problems, Late Acceptance Strategy in Hill-Climbing [17], Adaptive Randomized Descent Algorithm [18], Population based Local Search [19], Adaptive Acceptance Criterion algorithm [20], Effect of elite pool in hybrid population-based meta-heuristics [21] and many more.

However, almost approaches applied to benchmark datasets to investigate the performances of the algorithms, meanwhile; they are dealing with NP hard problem, therefore, the research move toward applying these algorithms over real world problem [22-26]. Recently Adaptive Acceptance Criterion algorithm (AAC) [20] investigated over different optimization problems and obtained good quality solution results outperformed different approaches results in the literature.
Therefore, this work motivates to propose a dynamic system for real exam timetabling problem by using AAC algorithms. The propose system, dynamically setting the days, timeslots, and constraints, which may can use for different exam timetabling problems.

This paper is organized as follow: section 2 present the problem description, section 3 present the methodology used to obtain feasible solution and its improvements, section 4 presents the implementation and discussion, and section 5 present the conclusion.

**PROBLEM DESCRIPTION**

Scheduling exams in Taibah University (TU) it is usually done manually subject to some constraints. Scheduling the exams in TU subject to university and departments constrains, where the university constrains consider as hard constrains and college/departments constrains consider as hard or soft relating to different college/department. Exams in TU schedule by three levels:

1. Scheduling general requirements exams.
2. Scheduling free electives exams.
3. Scheduling college or specialized exams.

Exams schedule is proposed based on the experience using white paper. The proposal is then advertise to the students and in turn they submit an objection in the event of exam conflict (if exist), and then another table is proposed and so on. These processes are divided into three levels, first, the general requirements exams are schedule, then secondly the free electives exams are schedule based on the first level, and finally the college or specialized exams are specified based on the second level. However, proposing the exam schedule in the third level take a lot of time and effort with increasing the possibility of students’ exams conflicts.

Exams usually scheduled based on even and odd term number, which differentiate between the regular and stumbling students. For instance, if there is a 4 year program includes 2 terms for each year, then we have a total of 8 levels with 4 odd level (i.e. level 1, 3, 5, and 7), and in case of the term is odd, then the odd levels are schedules first for regular students, later on the even levels schedule. Meanwhile, for regular students, each student have one day break between exams in each similar level exams.

In this work, the College of Computer Science and Engineering (CCSE) data are used as a test domain for proposing a dynamic real system for automated exam construction, which seek to optimize the students’ satisfaction for university exam timetabling problem. CCSE have three departments: Computer Science (CS) for male and female sections, Information System (IS) for male and female sections, and Computer Engineering (CE) for male section only but offered some courses to female sections. First three levels (level 1, level 2 and level 3) courses are unified between all departments (for more details description about the problem, please see Table1). The problem consists of:

- A set of courses, which is specified initially by the general and free electives courses.
- A set of courses from the college.
- A set of rooms, in which exams can take a place.
- A set of timeslots, in which exams can take a time.
- A set of students, who attends the course exam.

<table>
<thead>
<tr>
<th>Description</th>
<th>Male Section</th>
<th>Female Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of courses for the new students</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Number of outside college courses for the new students</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Number Of Courses</td>
<td>38</td>
<td>41</td>
</tr>
<tr>
<td>Number of Sections</td>
<td>102</td>
<td>139</td>
</tr>
<tr>
<td>Number of outside college courses</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>Number of CS courses</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>Number of IS courses</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Number of CE courses</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Number of courses that specify the exams from outside the college</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Number Of Courses</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Number Of Unified Courses between sections</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Number Of Unified Courses between departments</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Number Of Students</td>
<td>490</td>
<td>1100</td>
</tr>
</tbody>
</table>

All courses have to be scheduled into a number of timeslots and rooms. These data have four hard constrains (i.e. Hc1, Hc2, Hc3, and Hc4) and two soft constrains (i.e. Sc1 and Sc2), as follows:

**Hard Constrains**

Hc1: A set of courses, take a place in rooms with a specific timeslot.

Hc2: No student attends more than one exam at the same time.

Hc3: No student attends more than two exams at the same day.

Hc4: Room/s can accommodate the student number.

**Soft Constrains**

Sc1: Regular student should not have two consecutive exams at the same day or two consecutive days. In other words, should have a break between exam to another.

Sc2: Stumbling student should not have more than two exams in two consecutive days.
All hard constrains must be satisfied to obtain feasible solution, where the soft constrains are minimize as much as possible. Each soft constraint penalize as one to evaluate the exam timetable quality. The quality of exam timetable is measured based on the number of soft constraints violations (penalty cost).

**METHODOLOGY**

In general, there are certain methodology can be used to solve a problem. In this work, System Development Life Cycle (SDLC) [27] is used as the research methodology for proposing the exam timetabling system. SDLC consists of three phases as follows:

1. **Investigation phase**
   Investigation phase starts with clear understanding for the exam timetabling problem, outline the main issue of exam timetabling, and review the timetabling approaches in the literature.

2. **Implementation phase**
   Implementation phase concerns with the algorithm implementation and consist of three sub-phases as follows:
   - Initialization sub-phase, which starts by generating the conflict matrix to construct the initial solution for the problem.
   - Construction sub-phase, starts by satisfying the hard constrains on the initial solution to obtain feasible solution using sequential algorithm (graph coloring) [8, 10].
   - Improvement sub-phase, which try to improve the initial solution by minimizing the soft constrains while ensuring the feasibility using any algorithm.

3. **Comparison results phase**
   In the comparison phase, a comparison will be performed to evaluate the system performance, which compared to the previous timetable advertise (i.e. Term 1 year 2015-2016) and another data will be take into consideration for future work.

However, the investigation phase is described above, comparison phases as mention above it’s a future work to prove the effectiveness of the system over different datasets, where the implementation sub-phases need more explanation to understand how to system work as in 3.1, 3.2, and 3.3.

**Initialization sub-phase**

In this sub-phase, a matrix of events conflicts (which named as conflict or events/events) matrix (ECM) and stored the conflicts of events [28] is generated by initially generating:

- Student with event matrix (SEM) that stores the courses were students registered.
- Events with Rooms Matrix (ERM), which stores the suitable rooms for each even (exam).
- Room Capacity list, which store the room capacity of the students.
- Event with Students list, which store the number of student in each event (course).

**Procedure**

```plaintext
Procedure Events Conflict Matrix
For every student, retrieve the events chosen by the students (SEM);
For every event, calculate the students number in each event;
For every event i
   For every event j
      Calculate the number of students in common for every pair
         of events i and j where i,j \( \in \{1, \ldots, N\} \);
      Calculate the number of events in conflict;
      Calculate the number of students in each event;
   end For
end For
For every event:
   show the event index;
   show the number of students in each event;
   show the number of events in conflict;
For every event:
   show the conflicting events;
   show the number of events in conflict;
end For
end For
rearrange events/events matrix in descending order based on conflict
```

**Figure 1.** Pseudo code to generate Events Conflict Matrix (ECM)

**Construction sub-phase**

In this work, a constructive heuristic was proposed by Silva and Obit [28] is used. The heuristic works employing two neighborhood structure to add and remove the courses from the timetable by using tabu list in order to obtain feasible solution. These two neighborhoods are:

- Randomly select a course and randomly assign it to any feasible room and timeslot.
- Randomly select two courses and swap their rooms and timeslots while it is feasible.

The constructive heuristic has three steps (i.e. Highest Degree Heuristic Step, Local Search Step, and Tabu Search Step) as described in Figure 2 (for more details, please see [28]).

**Procedure**

```plaintext
Procedure Construction Algorithm
For counter=1 to number of events
   apply Step 1: highest degree heuristic;
end For
Repeat
   If infeasible timetable
      apply Step 2: neighbourhood structure.
      apply Step 3: tabu search.
   end If
Until feasible timetable
Return the feasible timetable
```

**Figure 2.** Pseudo code for the construction algorithm
Improvement sub-phase

Finally, the initial solution is improved by applying algorithm that may try to minimize the violation of soft constrains while ensuring the feasibility. Recently Adaptive Acceptance Criterion algorithm (AAC) [20] investigated over different optimization problems and obtained good quality solution results outperformed different approaches results in the literature, which motivates to use it in the improvement phase for the system.

AAC is an algorithm employs a non-parametric acceptance criterion that does not rely on user defined. AAC accepts a little worse solution based on stored estimated value. In AAC, an estimated value added to the threshold (when the search is idle) to increase the search exploration. The estimated value is generated based on the frequency of the solutions quality differences, which are stored in an array. Pseudo code for AAC is described in Figure 3 (please read [20] for more details).

**Implementation and Discussion**

In this work, the system implemented by using visual basic language on a PC with a processor of Intel core I5 - 3.2GHz and 4GB RAM. The system is described for the male section data. Figure 4 and Figure 5; show the main screen for the system.

![Main System Screen](image)

**Figure 4. Main System Screen**

Figure 4 shows, the main screen for the proposed examination system, where you can:
- Upload the file of student’s id’s numbers for each courses section or upload the file of the student’s id’s numbers by combining all sections for the similar course in one file. However, you can upload whole directory directly to the system file location (which are created in C: drive) as in Figure 5.
- Upload room files (boys and girls sections) that stores each room name with the student capacity number as in Figure 5.
- Parameter settings can be assign before creating the exam timetable, for example, exam default period as in Figure 6, exam default gap as in Figure 7, and sections constrains as in Figure 8.
- Choose the exam starting and ending days as in Figure 9.
Figure 5 shows that, the student in each course and rooms file are uploaded into the system with clarifying the path of it. In case of not uploading the correct files (courses and rooms), the system will not allow to use any of the system settings, otherwise, the settings will be available for use.

The system initialized with default exam time as two hours and the gap between each two exams is 30 minutes, however, the system allow the user to modify the initialized setting in Figure 6 and Figure 7. Figure 6 shows that, in case of the user checked for the exam default period setting, then the system will allow the user to choose different exam period to set as default period for all exams in hours. Also Figure 7 shows that, the system will allow changing the default gap time in minutes, and then set it as default gap time between two consecutive exams. Later on, the system allow the user to assign sections setting as in Figure 8, where in case of the user upload the files as courses section, then the system will allow to add a constraint to consider it as one course, or in case there are different plan in the departments and there is two different course number for the same course, then can combine it as one course.
Figure 9 shows that the system initialized with default staring exam time at 8AM and ending exam time 4PM in case of choosing exam days. The setting for the exams starting and ending days shows that you can:

- Choose the starting day date and ending day date.
- Exclude any day as a weekly holiday for all chosen days between starting and ending day.
- Exclude specific day as holiday between starting and ending day.
- Changing the default starting and ending exams time for all days.
- Build your files settings to construct the conflict matrix as in Figure 10.

Figure 10. Exam setting Confirmation
After confirming all settings as in figure 10, the courses/students conflict matrix is built as in Figure 11, then you can see the students/courses conflicts matrix in Figure 12, and for more clarification you can see students/courses in conflicts after optimize and exclude all conflicts duplications as in Figure 13. Please be noted, that the matrix is optional to follow to understand what happened, where you can directly move to the next step to construct feasible exam table as in Figure 14.
Figure 14. Constructing feasible timetable screen

Figure 14 show that, there are three constrains options before building the feasible exam timetable as follows:

- Exclude specific time from all days as in Figure 15.
- Exclude specific time from a specific day as in Figure 16.
- Exclude specific time or course for all days as in Figure 17.
Figure 15 shows that, the system allow excluding a specific time (e.g. Pray) for all days, where the timeslots reinitialized based on the new setting chosen. In addition, Figure 16 shows that it is possible to exclude a specific time for a specific day (e.g. 3/6/2018 Tuesday at 10:30-12:30). Furthermore, it’s allowed to choose a specific course for a specific day/time/room (e.g. 3/4/2018 Sunday at 13:00-15:00, CS323 course in room G86) as in Figure 17. After that, the system settings are specified and can move to the next step to build a feasible exam timetable as in Figure 18.
Figure 18 shows the room matrix, exam timetable, and assigned course list after building the feasible table and an improvement is done. Meanwhile, there is description for the exam timetable violations. For instance, in room matrix there is “1” in 3/4/2018 Sunday at 13:00-15:00, which reflect to the CS433 course exam in exam table and value of “1” in the course assigned list. However, the timetable description shows, that there is 0 students have three consecutive exams in one day, 0 students have two exams at same time in one day, and 85 students have two consecutive exams in one day.

In case of the user satisfied with exam timetable violations, then it is possible to be taken as it is, or rebuild the exam table to obtain a new exam timetable and son on. However, there is additional constrains can be apply to the exam timetable as in Figure 19, where can modify the maximum number of exams per day for all students and specify a number of days for the students as a break between exams if possible. These constrain, will reflect on the violation of the soft constrains and calculate it with the previous soft constrains and when we satisfy, we can stop the timetable optimizing and create the exam timetable.

Figure 19. Additional soft constrains for exam timetable

CONCLUSION

Exam timetable involve of allocating some students in courses to predefined rooms and timeslots subject to some constraints that must satisfy or others may satisfy. Several institution have their own way in generating the exam timetable that will suit their population size, which often may initially generating with some errors. In this paper, a dynamic real system is proposed by using the settings for day, time, and room dynamically, since it allows the used to input the resources available. Largest color degree first is used to generate a feasible timetable and Adaptive acceptance criteria algorithm is used for improvement. Pervious data for the timetable in CCSE- Taibah University is used as a test domain. The system show a good quality timetable that meets the regulations imposed while comparing it to the original timetable proposed that term, in which save time and money. Generally, the construction phase plays an important role in initializing the solution for improvement. In the proposed system, largest color degree first is used in the construction phase to obtain initial solution, where there are different graph coloring techniques can be used to obtain different solution structure to intensifying the search space that may help to obtain better solution in different universities data problem.
such as: largest Degree first, least Saturation degree first, largest Weighted degree first, and largest Enrolment first for obtaining different solutions structures. Also, the proposed system deals with the room capacity; however different university may needs rooms’ features for different exams. Which pose a future works, to enhance the system by intensifying the search space by multiple graph coloring techniques and add the rooms features to the system, then evaluate the effectiveness of the system in different datasets.

REFERENCES


