Scientific Research Publishing

University Timetable Generator Using Tabu Search

Tanzila Islam, Zunayed Shahriar, Mohammad Anower Perves, Monirul Hasan

Department of Computer Science and Engineering, Southeast University, Dhaka, Bangladesh
Email: tanzilamohita@gmail.com, zunayedshahriar@gmail.com, anowerperves@gmail.com, kmhasan@gmail.com

Abstract

This report is based on University Timetable Generator by using Tabu Search algorithm. It helps to generate a course schedule and an exam schedule for a University. Every university faces a different set of problem while preparing course schedule and exam schedule. There are lots of constraints while making a scheduler. And for this reason, students suffer much as well as faculties. This report is based on discussion about an automated timetable generator for a University by using Tabu Search algorithm. Tabu Search is a meta-heuristic procedure for solving optimization problems. Tabu Search deals with a sub-optimal initial solution. By analyzing the search space and averts inessential exploration, it optimists this solution and keeps the list of recently visited area in a Tabu list. This helps to solve these problems within a reasonable time and gives a feasible solution than any manual system. For a University, we have found that preparing exam schedule, course schedule, student assessment, room assignment with required resources are quite complex. But for all of them, we analyze that the Tabu Search technique is an essential method for getting a feasible solution. In this paper, we describe how Tabu Search works and how to get a feasible solution by using this algorithm.

Keywords

Tabu Search, Constraints, Scoring, Cost Benifit Analysis

1. Introduction

The timetabling problem is a problem of scheduling courses and exams which arises especially in higher educational institutions. While preparing timetable generator for an institution, lot of constraints arises. These constraints are divided into two types, one is hard constraints such as time conflict, room assignment, etc. and other one is soft con-

straints such as faculty preference. Here, hard constraints cannot be avoided for this timetabling problem. We will discuss about these constraints in Constraints section. These types of constraints may vary from one institution to another. The manual solution of this problem requires huge time and these types of solution are not feasible in most aspects. A lot of conflicts may find. An automated timetable generator can help to minimize the timetabling problem for any university.

There are some techniques to solve this timetabling problem. Among them, Constraint Satisfaction Problems (CSP), Tabu Search (TS), Simulated Annealing (SA), Genetic Algorithm (GA), Ant Colony Optimization (ACO) are the main approaches to solve this problem.

2. Research Objective

This paper is based on course scheduling and exam scheduling problem from the perspective of Southeast University. Students and faculties are facing lot of problems while preparing course schedule and exam schedule. Defining research objectives is a fundamental step which is necessary to reach the goal. The objectives of this research are:

1) To derive a suitable timetabling system for courses and exams with proper requirements.

2) To minimize constraints and find a feasible solution.

To achieve these objectives, the main step was finding the problems that our Faculties face while preparing these scheduling. Then gathering and analyzing the required information for the problem was done. As derived from the requirement analysis, the last task was to design the system and apply the Tabu Search algorithm for solving this timetabling problem.

2.1. Problem Overview

The timetabling problem arises in many real-life circumstances. A general timetabling problem includes many events like exam scheduling, course scheduling, meetings and so on. To satisfy all requirements within a limited time is responsible for these problems. There is a large amount of successful research on Timetabling Problem. For different timetabling problems, these researches have been done by various meta-heuristic approaches. University Timetable Generator involves with scheduling of courses and exams with satisfied faculty requirements, resources and room availability, time slots etc.

2.2. Constraints

The Timetabling Problem has some constraints. While preparing course or exam schedule, lot of constraints arise. These constraints can be divided into two types depends on priority. One is hard constraints which cannot be avoided and another is soft constraints. Constraints are varying from institution to institution but most of the time constraints are similar. In this paper, we describe about different types of constraints and find the feasible solution.

2.2.1. Hard Constraints

1) A faculty cannot take different classes at a time in the same room.

2) A faculty cannot take two classes at a time.

3) A student cannot appear in more than one class at a time.

4) A faculty cannot take a class if there is a distance problem from campus to campus.

5) A student should not have more than one exam at the same time.

6) A room cannot be assigned more than its capacity.

2.2.2. Soft Constraints

1) A student should not have more than one exam in a day.

2) We have to consider faculty preferences sometimes in some special cases.

2.3. Requirements

Major requirements for this research is to find out the problems which already have faced in an institution while preparing the course scheduling and exam scheduling. Those problems may vary from one institution to another. Therefore, finding the vision based solution should be easier.

2.4. Goal

"To build a University Timetable Generator by using Tabu Search algorithm which helps to prepare course schedule and exam schedule within a short time and gives a feasible solution."

2.5. Research Outline

In this paper, we present a simple but effective approach to solve this timetabling problem by using Tabu Search technique. In Section 3, we describe about some papers which has already published. There is also a brief description about Tabu Search, Fundamentals of Tabu Search and TS Algorithm. Requirement Analysis is given in Section 4. Solution Method and Scoring are in Section 5.

3. Literature Review

3.1. Related Paper Review

There are over hundreds of papers have been published about Timetable Generator. During the last fifty years, a lot of papers have been published in conferences and journals which are related to automate Timetable Generator. The first automated timetabling related research have been started from Gotlieb in 1963 [1]. A survey on automated timetabling is found on Schaerf (1999) [2]. It was about introducing into timetabling problems. In those circumstances, several applications have been developed with a great success. In 1967, Welsh and Powell used a graph coloring concept for reducing Timetabling Problem into the chromatic number problem which is also NP-Hard [3]. Dimopoulou and Milliotis have been using mathematical programming techniques to solve this problem [4].

Because of complexity, most of the researcher thinks about heuristic algorithm like Dave Corne, Emma Hart and Peter Ross have worked with Genetic Algorithms and timetabling [5], Mooney and Rardin with Tabu Search [6], Cooper and Kingstone with Simulated Annealing [7].

Some researchers have recently concentrated into Logic Programming because they want to solve this problem by a natural way of representing constraints. But there were some limitations with unsatisfied performance, which was pointed by Bartak [8].

Some articles describe practical case studies which are relatively compared to theoretical work on the Timetabling Problems. Michael W. Carter described a system named EXAMINE, which was applied in Universities of Toronto and Carleton [9]. Kendall and Hussin was described a Tabu Search meta-heuristic for ETP at University of Technology MARA [10]. Masri Ayob was described a formulation of an Examination Timetable Problem for University Kebangsaan Malaysia [11]. He proposed a useful objective function. There were some constraint based reasoning which has proven to be a productive research method for the researchers in different areas such as artificial intelligence, operational research and logic programming. There is a theoretical framework named Constraint Satisfaction Problem (Janssen, Jegou, Nouguier and Vilarem, 1989) which are solved by different versions of a backtracking search that deals with the binary.

Most of the papers describe an effective software implementation. In addition, the application of the method describes one or more test cases.

3.2. Tabu Search

Tabu Search (TS) is a meta-heuristic approach to explore the solution space beyond local-optimality. The main ideas of Tabu Search are based on the ideas proposed by Fred Glover (1977, 1986). Over more than the last 2 decades, over hundreds of papers have been represented application on Tabu Search. This method has become very popular for its capability to provide solutions very close to the best or near best optimally. TS provide a very flexible search behavior by using adaptive memory.

3.3. Fundamentals of Tabu Search

1) The way to exploit memory in TS is to classify a subset of the moves in a neighborhood as forbidden [12].

2) A neighborhood is built to identify adjacent solutions that can be reached from the current solution [13].

3) The classification, particularly depends on the recency that certain move or solution components, called attributes, have participated in generating past solutions [12].

4) A tabu list which records forbidden moves are referred as tabu moves [14].

5) When a tabu move has a more satisfied evaluation where it would result in a solution better than any visited so far, then its tabu classification may be overridden [12].

3.4. TS Algorithm

Pseudocode for Tabu Search [15].

Input : TabuList$_{size}$
Output : S$_{best}$

1. S$_{best}$ ← ConstructInitialSolution();

2. TabuList ← ϕ;

3. while ¬ StopCondition() do

4.   CandidateList ← ϕ;

5.   for S$_{candidate}$ ∈ Sbest$_{neighbourhood}$ do

6.   if ¬ ContainsAnyFeatures (S$_{candidate}$, TabuList) then;

7.   CandidateList ← S$_{candidate}$;

8.   end

9.   end

10.   S$_{candidate}$ ← LocateBestCandidate(CandidateList);

11.   if Cost(S$_{candidate}$) ≤ Cost(S$_{best}$) then

12.   S$_{best}$ ← S$_{candidate}$;

13.   TabuList ← FeatureDifferences(S$_{candidate}$, S$_{best}$);

14.   while TabuList > TabuList$_{size}$ do

15.   DeleteFeature(TabuList);

16.   end

17.   end

18. end

19. S$_{best}$;

4. Requirement Analysis

This requirement analysis is based on Southeast University. There are two types of program in this University, one is Undergraduate Program and another is Postgraduate Program. School of Science & Engineering, School of Business Studies, School of Arts & Social Science and Department of English are belongs to Undergraduate Program. There are six departments under School of Science & Engineering. These are Department of Computer Science & Engineering (CSE), Department of Information & Communication Engineering (ICE), Department of Electrical & Electronics Engineering (EEE), Department of Pharmacy, Department of Textile Engineering and Department of Architecture. Bachelor of Business Administration (BBA) belongs to School of Business Studies. School of Arts & Social Science consists of two departments: The Bachelor of Laws (LLB) and Department of Economics. Masters in Business Administration (MBA), Master of Laws (LLM) and Masters in English are belong to Postgraduate Program.

From **Tables 1-3**, we have found different types of criteria from all the departments.

Table 1. Undergraduate programs (school of science & engineering).

Analysis	CSE	ICE	EEE	Pharmacy	Textile	Arch.
Starting Time	8.30 am	8.30 am	8.30 am	8.30 am	8.30 am	8.30 am
Ending Time	5.20 pm	5.20 pm	5.20 pm	6.00 pm	5.20 pm	5.20 pm
Working Days	4	4	4	6	5	6
Pattern	ST/MW	ST/MW	ST/MW	No	No	No
Faculty Preference	High	High	High	Medium	Medium	Low
Current Routine	Software	Software	Software	Manually	Manually	Manually
Theory Cr. Hour	3 h	3 h	3 h	3 h	3 h	3 h
Lab Cr. Hour	1.5 h	1.5 h	1.5 h	4 h	2.5 h	16 h
Break Time	10 min	10 min	10 min	30 min	10 min	10 min
Slots Per Day	6	6	6	8	6	6

Table 2. Undergraduate programs (school of business studies, school of arts & social science and department of English).

Analysis	BBA	LLB	English	Economics
Starting Time	8.00 am	8.30 am	8.30 am	8.30 am
Ending Time	6.20 pm	5.20 pm	5.20 pm	5.20 pm
Working Days	4	7	4	4
Pattern	ST/MW	No	No	ST/MW
Faculty Preference	High	Low	Medium	Medium
Current Routine	Manually	Manually	Manually	Manually
Theory Cr. Hour	3 h	3 h	3 h	3 h
Break Time	10 min	10 min	10 min	10 min
Slots Per Day	7	6	6	6

Table 3. Postgraduate programs.

Analysis	MBA	MBA	LLM	MA in English
Programs	Regular	Executive	-	-
Duration	16 months	2 year	1 year	1 year
Starting Time	6.30 pm	6.30 pm	8.30 am	3.00 pm
Ending Time	8.00 pm	8.00 am	7.00 pm	8.30 pm
Working Days	2	2	2	3
Faculty Preference	High	High	Low	High
Current Routine	Manually	Manually	Manually	Manually
Slots Per Day	1	1	1	3

Most of the criteria are similar. For example, most of the departments maintaining their routine manually, starting their classes at the same time etc.

5. Solution Method

To solve this entire University Timetable Generator problem, we have to introduce scoring where we will try to make a best routine by using some score. We will see how scoring will help us to solve this problem using scoring.

Scoring

From Figure 1 we can see that, scoring is helping us to get a better score routine. Now we will see how the score works behind this.

For Example:

Assume that, there is a university named "Example University" contain only one department. Let's name it Department of Computer Science and Engineering at Example University. Guess that there are 4 (four) teachers and 50 (fifty) students with 10 (ten) courses.

So,

Department: 1

Teachers: 4

Students: 50

Courses: 10

Teachers Preference: high

Setting up the teacher preferences (Ex: Seniority based order):

Teacher, A = 4000

Teacher, B = 300

Teacher, C = 200

Teacher, D = 100

if we break the promises to the Teacher A, we will get −4000 score for him/her and if the other Teachers are fine then we will get +600 (=300 + 200 + 100) for all of them. We are handling Teacher preferences like this.

While a student taking multiple courses, if any of his courses scheduled at same time at the routine, then the routine will give us a big negative value (−10,000). By assigning this type of negative values we will generate a routine with a total penalty score, then we will throw this to the Tabu Window (Figure 1) using Tabu Search. Every time it will compare to the routine stored in the Tabu Window and if the current routine is better than any other routine in the Tabu Window then it will store this into the Tabu Window and will throw out the last routine from the window. For Example: There are 10 (ten) routine stored in the Tabu Window, sorted against the score

−4950 −5460 −5570 −9520 −11,290 −12,456 −17,529 −21,220 −22,458 −25,798

Now we have generated a routine with score −6570. Then the last routine (−25,798) from the Window will be removed and we will store our current routine in the Tabu Window. Then the window will looks like-

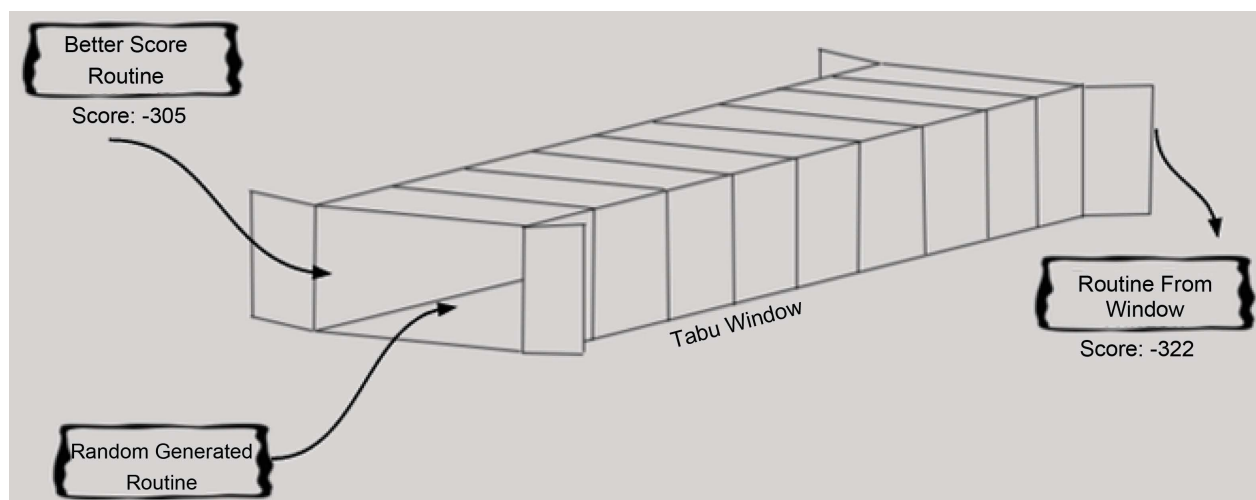−4950 −5460 −5570 −6570 −9520 −11,290 −12,456 −17,529 −21,220 −22,458

Figure 1. Tabu window.

6. Cost Benefit Analysis

The University Timetable Generator will cost effective work. It will save some valuable time our departmental Coordinators, section officers, program officers, teachers. It will reduce student class conflicts and exam conflicts problems.

Positive and Negative Sides

Manually semester routine creating is a time consuming work. A departmental Coordinator is always investing his/her a lot of time for this. Sometime other teachers and section officers are working along with a departmental Coordinator to generate a semester routine. A University Timetable Generator will reduce time and resources. Students and teachers will get best possible routine for a particular semester through this generator.

Though it will generate a routine automatically, there will not be possible to keep all preferences all time. Sometimes it can happen that some teachers are not getting their schedule according to their preferences. If we use multiple campuses for a particular department, then students have to travel a lot and teachers too.

7. Conclusion

This research is a package for scheduling courses and exam of a University. This package will help to handle a timetabling problem by satisfying all the requirements. Tabu Search algorithm obtains a high quality solution within a short time. If it can't manage the best optimal solution, it will then provide a solution which is optimizing the nearest to the best. Through this research, using Tabu Search algorithm a University Timetable Generator can be implemented so easily.

Acknowledgements

References

[1] Sandhu, K.S. (2003) Automating Class Schedule Generation in the Context of a University Timetabling Information System.
https://www120.secure.griffith.edu.au/rch/items/9d83345c-4327-3a99-24ff-b42bb508c960/1/

[2] Schaerf, A. (1999) A Survey of Automated Timetabling. *Artificial Intelligence*, **13**, 87-127.

[3] Welsh, D.J.A. and Powell, M.B. (1967) An Upper Bound for the Chromatic Number of a Graph and Its Application to Timetabling Problems. *The Computer Journal*, **10**, 85-86.
https://doi.org/10.1093/comjnl/10.1.85

[4] Dimopolou, P.M.M. (2001) Implentation of a University Course and Examination Timetabling System. *European Journal of Operational Research*, **130**, 202-213.
https://doi.org/10.1016/S0377-2217(00)00052-7

[5] Ross, P., Corne, D. and Hart, E. (2003) Genetic Algorithms and Timetabling. Springer Berlin Heidelberg, 755-771.

[6] Mooney, E.L. and Rardin, R.L. (1993) Tabu Search for a Class of Scheduling Problems. *Annals of Operations Research*, **41**, 253-278. https://doi.org/10.1007/BF02023077

[7] Cooper, T.B. and Kingston, J.H. (1993) The Solution of Real Instances of the Timetabling Problem. *The Computer Journal*, **36**, 645-653. https://doi.org/10.1093/comjnl/36.7.645

[8] Muller, T. and Bartak, R. (2002) Interactive Timetabling: Concepts, Techniques and Practical Results. Master Thesis, KTIML MFF UK, Prague, September 2001, 58-72.

[9] Carter, M.W., Laporte, G. and Lee, S.Y. (1996) Examination Timetabling: Algorithmic Strategies and Applications. *The Journal of the Operational Research Society*, **47**, 373-383.
https://doi.org/10.1057/jors.1996.37

[10] Kendall, G. and Hussin, N.M. (2005) A Tabu Search Hyper-Heuristic Approach to the Examination Timetabling Problem at the Mara University of Technology. Series Volume: 3616, Springer Berlin Heidelberg, 270-293.

[11] Ayob, M., Malik, A.M.A. and Kendall, G. (2007) Solving a Practical Examination Timetabling Problem: A Case Study. Series Volume: 4707, Springer Berlin Heidelberg, 611-624.

[12] Glover, F., Kelly, J.P. and Laguna, M. (1995) Genetic Algorithms and Tabu Search: Hybrids for Optimization. *Computers and Operations Research*, **22**, 111-134.
https://doi.org/10.1016/0305-0548(93)E0023-M

[13] Reeves, C.R. (1993) Modern Heuristic Techniques for Combinatorial Problems. John Wiley & Sons, Inc., New York, 320.

[14] Hillier, F.S. and Leiberman, G.J. (2005) Introduction to Operations Research. Tata McGraw Hill.

[15] Brownlee, J. (2012) Clever Algorithms Nature-Inspired Programming Recipes. lulu.com.

Scientific Research Publishing

Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.
A wide selection of journals (inclusive of 9 subjects, more than 200 journals)
Providing 24-hour high-quality service
User-friendly online submission system
Fair and swift peer-review system
Efficient typesetting and proofreading procedure
Display of the result of downloads and visits, as well as the number of cited articles
Maximum dissemination of your research work

Submit your manuscript at: http://papersubmission.scirp.org/
Or contact jcc@scirp.org