International Journal of Scientific & Engineering Research, Volume 5, Issue 10, October-2014
ISSN 2229-5518

University Examination Timetabling Using Tabu Search

Hamid D. Lawal[1], Ibrahim A. Adeyanju[2,*], Elijah O. Omidiora[2], Oladiran T. Arulogun[2] and Oluyinka I. Omotosho[2]

Abstract— University Examination Timetabling deals with scheduling examinations for students while satisfying specified constraints. Such constraints include avoiding overlap of courses offered by same students, ensuring fair spread of courses offered by a class and allocating suitable venues. This paper discusses an automated examination timetabling system for universities based on Tabu Search, a meta-heuristic technique. Tabu search starts with random generation of an initial solution which is typically sub-optimal. It then gradually optimises this solution by exploring the search space but avoids unnecessary exploration by keeping a list of recently visited areas in a Tabu list. Three versions of the timetabling system (labelled Sys_A, Sys_B & Sys_C), having varying penalties for violating constraints specified as hard or soft, were evaluated. Sys_A used equal penalty on all constraints, Sys_B used equal penalty on all hard constraints and a lower equal penalty on all soft constraints while Sys_C used different penalties for hard constraints depending on their perceived importance and the lowest penalty on soft constraints. The experimental dataset consisted of 153 courses with varying class sizes for a total of 5550 students to be scheduled within 25 days using 15 venues of different capacities. The data was obtained from the Faculty of Engineering and Technology, LAUTECH, Nigeria. The time taken to generate a timetable within 1000 maximum iterations and weighted relative error of generated timetables were used as evaluation metrics. The least error (best result) was obtained with Sys_B, having equal penalty on all hard constraints and a lower equal penalty on all soft constraints though with a second best (lower) simulation time.

Index Terms— Constraints satisfaction, Hard and soft constraints, Scheduling, Tabu search, Timetabling , University examination

◆

1 Introduction

Timetabling is a combinatorial problem that commonly arises in higher institutions [1] and essentially concerned with scheduling a certain number of examinations in a given number of time slots in such a way that no student is having more than one examination at a time. Timetabling problems are in the set of NP-hard problems [2] [3]. Assigning examinations to days and timeslots within the day are also subject to constraints. These constraints are divided into two types: hard constraints and soft constraints; and may vary from one institution to another. The hard constraints are the compulsory ones that cannot be violated while soft constraints are necessary to improve the quality of a timetable but not compulsory [4]. The manual solution of the timetabling problem usually requires several days of work and the solution obtained may be unsatisfactory in some respect; therefore, considerable attention has been devoted to automated timetabling [5] [6].

Genetic Algorithm, Simulated Annealing [7] [8], Memetic Algorithm, Tabu Search and Ant Colony Optimisation [9] are among the main approaches for solving the timetabling problem intelligently. In this paper, we propose a simple but effective approach to solving timetabling problem using Tabu search. Section 2 discusses some related work while Section 3 describes the Tabu search approach to Examination Timetabling. Our experimental methodology and discussion of results are given in Sections 4 and 5. The paper is concluded in Section 6 with pointers to preferable extensions to the current work.

2 Related Work

Examinations Timetabling Problem (ETP) is the problem of assigning courses to be examined, candidates and examination rooms to time slots while satisfying a set of constraints. It is an important issue in higher institutions and is known to be a highly constraint-based problem [10] [11]. This problem is typically characterized as a constraint satisfaction problem that is complex in nature and very difficult to solve. To overcome this problem, higher institutions need an automated examination timetabling scheduler that is robust, flexible (can accommodate new courses and venues), conflict-free (satisfies virtually all the specified constraints) and generates a time table within few minutes.

Ant Colony Optimisation and Simulated Annealing were used by [12] and [13] respectively while [14] [15] proposed Tabu Search as approaches to solving ETP. Though the Tabu search approach is similar to the one used in this paper, hard and soft constraints were not differentiated in previous works. Chu & Fang [16] worked on the Genetic Algorithm (GA) and Tabu Search in timetable scheduling and compared the performance of these two techniques based on the quality of the exam timetable and the time spent in producing the timetable. Their results show that TS produced better timetable than GA, but search time spent in TS is less than that of GA while GA produces several different near optimal solutions simultaneously.

* *Corresponding Author: I.A. Adeyanju. Email: iaadeyanju@lautech.edu.ng*

1. *Department of Computer Science, Federal Training Centre, Ilorin, Kwara State Nigeria.* ***Email:*** *hamidlawal2@gmail.com*
2. *Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Oyo state,* Nigeria

Emails: *{iaadeyanju | eoomidiora | otarulogun | oiomotosho}@lautech.edu.ng*

ISSN 2229-5518

Oyeleye et al [7] worked on the performance evaluation of simulated annealing (SA) and GA in solving examination timetabling problem. He evaluated the performance of the two approaches based on the computing resources used in terms of software complexity measures. Their results indicate that GA consumed very high computing resources but with high optimality as compared to SA. They then proposed the development of a hybrid algorithm consisting of both GA and SA to reduce the weaknesses and combine their strengths to produce solutions more optimal than those produced individually.

3. TABU SEARCH

Tabu Search (TS) is a global heuristic technique which tries to avoid falling into local optima by creating a special list called tabu. Any solution which has been recently selected is put into a tabu list so that it becomes 'taboo' for a short period of time, depending on the length of the list. This minimizes the chance of cycling in the same solution, and therefore creates more chances of improvement by moving into unexplored areas of the search space [17] [18]. However, the major advantage of TS is to use its memorized ability to prevent searching areas previously visited. Therefore, it is easier to escape from local optimum and approach the global or near global optimum solutions in a short time. DiGaspero, McCollum and Schaerf [19] carried out a valuable investigation on a family of Tabu search based techniques whose neighbourhoods concerned those which contribute to the violations of hard or soft constraints. The length of the tabu list is dynamic and the cost function so adaptively set during the search. The authors experimentally demonstrated that the adaptive cost function and the effective selection of neighbourhoods concerning the violations were key features of the approach.

Tabu search is a meta-heuristic that has successfully been applied to find good feasible solutions for hard optimization problems [20] [21]. In general, it can be described as a neighbourhood search method incorporating techniques for escaping local optima and avoid cycling [22]. TS comprises six major components: Current starting solution, Neighbourhood search, Move, Evaluation, Tabu list and Aspiration criteria. The number of iterations during which this move is kept in the tabu list may follow different strategies such as keeping the dimension of the tabu list fixed and limited, assigning a value for the tabu tenure equal for all moves or assigning different values of permanence in the tabu list depending on historical record. Tabus are one of the distinctive elements of TS and are used to prevent cycling when moving away from local optima through non-improving moves. The key realization here is that when this situation occurs, something needs to be done to prevent the search from tracing back its steps to the source [23]. This is achieved by declaring tabu (disallowing) moves that reverse the effect of recent moves. Tabus are stored in a memory of the search (the tabu list) and usually only a fixed and fairly limited quality of information is recorded. In any given context, there are several possibilities regarding the specific information that is recorded. Figure 1 shows the flowchart for a standard tabu search.

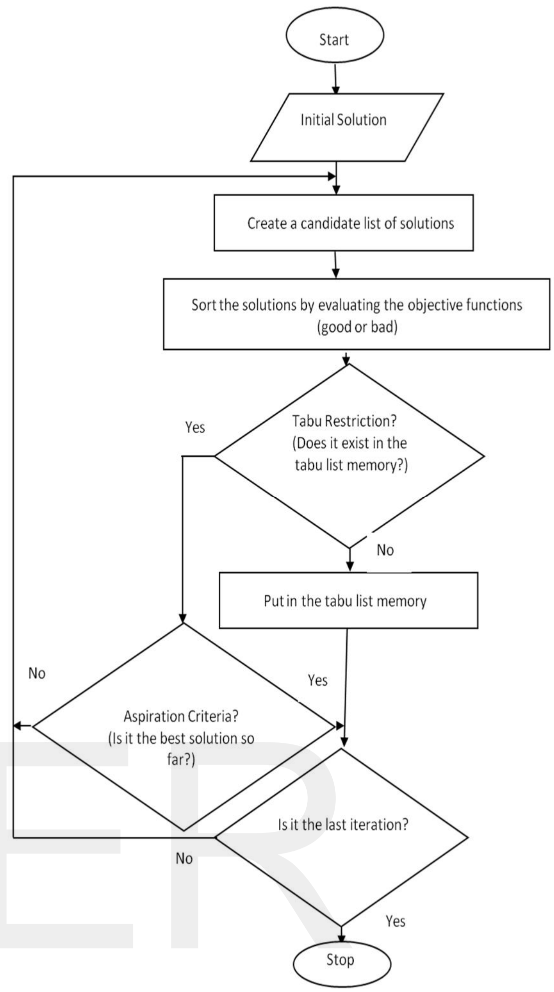

Figure 1: Flowchart of a Standard Tabu Search [24]

The first two basic elements of any TS heuristic are the definition of its search space and its neighbourhood structure. The search space of a TS heuristic is simply the space of all possible solutions that can be considered (visited) during the search. Closely linked to the definition of the search space is the neighbourhood structure. At each iteration of TS, the local transformations that can be applied to the current solution, denoted S, define a set of neighbouring solutions in the search space; denoted N(S) (the neighbourhood of S) is a subset of the search space defined by: N(S) = {solution obtained by applying a single local transformation to S}. While central to TS, tabus are sometimes too powerful; they may prohibit attractive moves, even when there is no danger of cycling, or they may lead to an overall stagnation process. It is thus necessary to use algorithmic devices that will allow one to revoke (cancel) tabus. These are called aspiration criteria .The simplest and most commonly used aspiration criterion found in almost all TS implementations consist in allowing a move, even if it is tabu, if it results in a solution with an objective value better than that of the current best-known solution (since the new solution has obviously not been previously visited). Much more complicated aspiration criteria have proposed and successfully implement [17], but they are rarely used. The key rule in this respect is that if cycling cannot occur, tabus can be disregarded.

The objective function, in which each individual solution appears with its corresponding weight, is also important. The objective function indicates how much each variable contributes to the value to be optimized in the problem. In theory, the search could go on forever, unless the optimal value of the problem at hand is known beforehand. However in practice, the search has to be stopped at some time point. The most commonly used stopping criteria in Tabu Search are: after a fixed number of iterations (or a fixed amount of CPU time), after some number of iterations without an improvement in the objective function value (the criterion used in most implements) or when the objective reaches a pre-specified threshold value. In complex tabu schemes, the search is usually stopped after completing a sequence of phase, the duration of each phase being determined by one of the stopping criteria [14]).

4. Evaluation Methodology

Our evaluation data was limited to a single faculty (Faculty of Engineering and Technology) in a university (Ladoke Akintola University of Technology, Ogbomoso, Nigeria) for ease of result analysis. The faculty consists of seven (7) departments with about 5,550 students, 150 courses, and 15 examination venues with a maximum of 25 days for the duration of the entire exam. The data was obtained from the Faculty of Engineering and Technology, LAUTECH, Nigeria. The only soft constraint specified was that no student should have two exams consecutively while hard constraints were specified as follows:

(i) No student should sit for more than one examination at any one time.
(ii) The capacity of rooms must match with student size.
(iii) No candidate can be assigned to more than one examination at the same time.
(iv) No exams should be scheduled between 12 – 3 pm on Friday due to Muslims' Friday prayer.

The developed examination timetabling system was also based on the following assumptions:

- An examination should be scheduled based on room capacity.
- All examinations are three hours long. In practice, some examinations are 2½ hours and some are 3½hours long but can be considered to fit within the 3 hours timeslots.
- Once assignment of courses to rooms and timeslots is done, a division of courses in lecture theatres into different groupings can be finished up manually.
- Walking distance between examination rooms is irrelevant, as there is an interval of at least one and half hours between examinations.

The performance of the Tabu search based timetabling system was evaluated using error rate and simulation time. Three versions of the timetabling system (labelled Sys_A, Sys_B & Sys_C), having varying penalties for violating constraints specified as hard or soft, were evaluated. Table 1 shows the different weight of penalties of hard and soft constraints considered in this work across three different constraint configurations. Higher weights mean that the constraint is more important. The first four constraints (class too small, room clash, exam clash, and Friday prayer times) are hard constraints which must be satisfied while the fifth (consecutive exams) is a soft constraint which we want to satisfy but not compulsorily. The three constraint configurations were tested to indicate how important it is to differentiate between hard and soft constraints as well as given some hard constraint priority over others. In the first configuration denoted as "Sys_A", each constraint is given the same weight of 10 irrespective of whether they are hard or soft. The second configuration, "Sys_B", uses the same weight of 10 for the four hard constraints but a smaller weight of 1 for the soft constraint. In the third configuration labelled as "Sys_C", hard constraints are given different weights from 4 to 10 depending on their level of importance but the weight on the soft constraint is lower at 1.

Table 1: Timetabling Constraint Penalty Weights

	Configurations		
Constraints	Sys_A	Sys_B	Sys_C
Venue Capacity Small	10	10	6
Room Clash	10	10	8
Exam Clash	10	10	10
FridayPrayer	10	10	4
Consecutive Exam	10	1	1

5. Experimental Results

The results of experiments with the Tabu search-based examination timetabling system are shown in Table 2 using different constraint errors, fitness function and simulation time with respect to the three constraint configurations explained in Section 4. Analysis of Table 2 shows that configuration "Sys_B" generated the timetable with the best quality compared to the other configurations. This is because it has identical constraint errors and fitness function to the "Sys_C" configuration with both lower than "Sys_A", but the time used to generate the timetable for "Sys_B" was average (4.2 minutes). This shows that differentiating between hard and soft constraints is important but differentiating between several hard constraints does not make any significant difference.

The results also show that the Exam clash constraint was the most difficult hard constraint to satisfy for Tabu search having a 0.7% error as compared to 0% for other hard constraints, based on the results of the "Sys_A" configuration. Venue capacity, Room clash and Friday constraints were the easiest hard constraints to satisfy as they had 0% for the three configurations. The only soft constraint investigated (consecutive exam) had a small error (0.7%) for two configurations (Sys_B and Sys_C) but was satisfied by the "Sys_A" configuration. The soft constraint was satisfied for the

"Sys_A" configuration since it did not differentiate between hard and soft constraints.

Table 2: Experimental Results (Error rates, Objective Function and Simulation Time)

Constraints	*Configurations*		
	Sys_A	Sys_B	Sys_C
Venue Capacity Error	0%	0%	0%
Consecutive Exam Error	0%	0.7%	0.7%
Exam Clash Error	0.7%	0%	0%
Friday Error	0%	0%	0%
Room Clash Error	0%	0%	0%
Objective Function	10	1	1
Simulation time (mins)	2.9	4.2	31.4

6. Conclusion

Tabu search has been shown to be effective in solving examination timetabling problem. Empirical results also indicate that better timetables can be produced when hard and soft constraints are differentiated. This work, when put to practice, can help universities to drastically minimize the time spent for preparing examination timetable manually, which usually requires several days of work with the final solution still sometimes unsatisfactory as it is an NP hard problem.

Future work includes scaling up our experiment data to the entire university and formulating all possible constraints. It will also be interesting to compare the performance of Tabu search to other state-of-the-art techniques for university examination time tabling.

References

[1] D.C. Wood, "A system for computing university examination timetables", The Computer Journal, vol. 11, no. 1, pp. 41-47, 1968.

[2] S. Abdennadher and M. Marte, "University course timetabling using constraint handling rules", Journal of Applied Artificial Intelligence, vol. 14, no. 4, pp. 311 – 326, 2000.

[3] Z.N. Azimi, ''Hybrid heuristics for Examination Timetabling problem'', Applied Mathematics and Computation, vol. 163, no. 15, pp. 705 – 733, 2005.

[4] K.P. Dahal, K.C. Tan and P.I. Cowling, Evolutionary Scheduling, Springer press, Berlin Heidelberg, Germany, 2007.

[5] D. deWerra, "An introduction to timetabling". European Journal of Operational Research, vol. 19, pp. 151-162, 1985.

[6] V. Lotfi and R. Cerveny, "A final exam-scheduling package", The Journal of the Operational Research Society, vol. 42, no. 3, pp. 205-216, 1991.

[7] C.A. Oyeleye, E.O. Omidiora, S.O. Olabiyisi and J.B. Oladosu, "Performance Evaluation of Simulated Annealing and Genetic Algorithm in Solving Examination Timetabling Problem", Scientific Research and Essays, vol. 7, no. 17, pp. 1727-1733, 2012.

[8] K. A. Dowsland, "Simulated annealing solutions for multi-objective scheduling and timetabling", Modern heuristic search methods, New York: Wiley, pp. 155-166, 1996.

[9] L. DiGaspero, B. McCollum and A. Schaer, "Curriculum-based course timetabling track", Proceedings of the 14th RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion, Rome, Italy, 2007

[10] [M. Eley, "Ant Algorithms for the Exam Timetabling Problem", Sixth International Conference on the Practice and Theory of Automated Timetabling (PATAT'06), 2006.

[11] M.W. Carter and D.G. Johnson, "Extended partition initialization in examination timetabling". The Journal of the Operational Research Society, vol. 52, pp. 538-544, 2001.

[12] E.K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper-heuristic for educational timetabling problems", European Journal of Operational Research, vol. 176, pp. 177-192, 2007.

[13] K.A. Dowsland and J. Thompson, "Ant colony optimization for the examination scheduling problem". The Journal of Operational Research Society, vol. 56, pp. 426-438, 2005.

[14] J. Thompson and K. Dowsland, "Variants of simulated annealing for the examination timetabling problem". Annals of Operations Research, vol. 63, pp. 105-128, 1996.

[15] A. Hertz, "Tabu Search for Large Scale Timetabling Problems", European Journal of Operation Research, vol. 54, no. 1, pp. 39-47, 1991.

[16] D. Clark, "Examinations Scheduling by Tabu Search", Australian Society Operations Research Bulletin, vol. 12, pp. 5-9, 1993.

[17] S.C. Chu and H.L. Fang, "Genetic Algorithms vs. Tabu search in Timetable scheduling", Third International Conference on Knowledge-Based Intelligent Engineering System, Australia, pp. 492-95, 1999.

[18] D. DeWerra and A. Hertz, "Tabu Search Techniques: A tutorial and an application to neural networks", OR Spectrum, pp. 131-141, 1989.

[19] G. M. White, B.S. Xie, and S. Zonjic, "Using tabu search with longer-term memory and relaxation to create examination timetables", European Journal of Operational Research, vol. 153, no.16, pp. 80-91, 2004.

[20] N. Hussin,"Tabu search based hyper-heuristic approaches for examination timetabling", Ph.D. thesis, Department of Computer Science, University of Nottingham, 2005.

[21] S. Abdullah, S. Ahmadi, E.K. Burke, M. Dror, and B. McCollum, "A tabu-based large neighbourhood search methodology for the capacitated examination timetabling problem". Journal of the Operational Research Society, vol. 58, pp. 1494-1502, 2007.

[22] F. Glover and M. Laguna, Tabu search, Norwell MA: Kluwer Academic Publisher, 1997.

[23] F. Glover, E. Taillard , M. Laguna and D. deWerra, "Tabu Search", Annals of Operations Research, vol. 41, 1993.

[24] S. Muhammed, Examination Timetable Scheduling Using Tabu Search, MSc Thesis, University of Utara, Malaysia, 2008.

Author's Biographies

Hamid D. Lawal is an Academic Staff at Department of Computer Science, Federal Training Centre, Ilorin, Kwara State Nigeria. This research work was done as part of his Masters of Technology (M.Tech.) thesis at the Ladoke Akintola University of Technology, Ogbomoso, Oyo state, Nigeria.

Dr. I.A. Adeyanju, Prof. E.O. Omidiora, Dr. O.T. Arulogun and Mrs O.I. Omotosho are academics (Lecturer 1, Professor, Senior Lecturer and Lecturer 1, respectively) at the Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Oyo state, Nigeria where they teach and conduct their research.