

IMPROVING EXAM TIMETABLING SOLUTION USING TABU SEARCH

AHAMAD TAJUDIN KHADER, ANG SIEW SEE
School of Computer Sciences,
Universiti Sains Malaysia, Penang, Malaysia
tajudin@cs.usm.my, siewsee@hotmail.com

ABSTRACT

A feasible exam timetable is generated using a method based on constraint satisfaction and heuristics. We investigate the usage of tabu search to further improve the quality of the exam timetable. Different length of short term tabu list and the long term tabu list is examined. Short term tabu list without long term tabu list and vice versa is also tested. Different search iteration based on maximum null iteration and maximum tabu relaxation is also considered. Experiments are carried out on an actual dataset from Universiti Sains Malaysia. Results from these experiments show the relative significance of long term tabu list relative to short term tabu list for this dataset.

Keywords: exam timetabling, tabu search, meta heuristics

1 INTRODUCTION

A feasible solution for an exam timetabling problem is a solution that satisfies all the hard constraints while trying to maximize the satisfaction of the soft constraints. Given a feasible exam timetabling solution we look for ways to improve the quality of the solution. In other words, we like to increase the satisfaction of soft constraints while maintaining the satisfaction of hard constraints.

Recent research work based on the tabu search technique shows its capabilities to produce an optimal solution with the usage of the structured memory by A. Scharef & L. Di Gaspero [1], George M. White and Bill S. Xie [2], and L. Paquete and T. Stützle [3].

In this paper, we present our work in repairing a feasible exam timetabling solution using Tabu Search. We focus on the usage of memory structure to improve the quality of the solution. The memory structure we use includes recency-based short term memory and frequency-based long term memory. The formation of parameter used to tune the memory structure is very important to the success of tabu search technique application as suggested in Michel Gendreau [4].

Hence, experiments are conducted to investigate the effect of memory structure over the quality of solution produced. We investigated the usage of tabu search to improve the quality of the solution. Attempts were made using a combination of different length short term tabu list with different length long term tabu list. The search was also varied by using several different combinations of maximum null iteration and maximum tabu relaxation.

2 EXAM TIMETABLING

The exam timetabling problem is the assignment of exams to timeslots while satisfying a set of constraints. Improving a feasible exam timetabling is by further trying to minimize the violation of soft constraints while maintaining the hard constraints are being satisfied. In an optimal case, the solution will have both hard and soft constraints satisfied.

Hard constraints are constraints that must be satisfied in order to create a feasible solution. The hard constraints that we take into consideration for our problem includes:

No clashing - not to have more than one exam at one timeslot for a same student. (i.e. a student should not take more than one exam at one particular timeslot.)

Exam availability - some exams must happen together at the same time. Due to the nature of the subjects, certain exams are required to be held at one same timeslot. (i.e. exams with more than one subject codes, exams with different group of students, the lecturer likes the exams to be held at the same time). These exams that have to be scheduled in the same time slot are referred to as *concurrent exams* in this paper.

Room capacity – Numbers of students sitting for all exams in one particular timeslot must not exceed the capacity at that timeslot.

Timeslot constraint – some exams are not allow to be held on a predefined range of timeslots.

In this research, the soft constraints that we wish to satisfy includes :

No consecutive exams – For each student, one exam in a day is always preferred.

Spacing between exams - More spacing between exams should be allocated to maximize student's exams preparation time.

3 RELATED WORK

Andrea Schaerf and Luca Di Gaspero [1], reduced the exam timetabling problem into a graph colouring problem. Exams are assigned to nodes and colours are used to represent the timeslots. An undirected edge between two nodes u and v is assumed if at least one student is enrolled in both the exams u and v . An edge-weight function $st : E \rightarrow N$ is then used to represent the number of student enrolled in each exam. Two types of neighbourhood are used. The first one called violation list (VL) contain nodes that are involved in at least one violation either hard or soft constraint and the second tabu list is called HVL that contain only nodes that are involved in hard constraints. During search, nodes are selected by exploration of HVL when there are some hard constraints and resort to VL in any iteration when HVL is empty. Nodes that are not in the list VL or HVL will never be examined. Once best solution is determined, new colour is assign to the selected node u that yields the smallest cost function. For initial solution, Andrea Schaerf and Luca Di Gaspero [1] experimentally observed that a good initial state is important as it will saves time and more complete exploration of the search space. They use a greedy algorithm to builds the initial solution.

George M. White and Bill S. Xie [2] developed an algorithm called OTTABU system adapting Tabu Search technique for their exam timetabling problem. OTTABU system uses both recency-based short term memory and frequency-based longer term memory to improve the quality of an exam timetabling solution. They showed that the combination use of frequency based longer-term memory and tabu relaxation technique helps to optimize the exam timetabling problem.

George M. White and Bill S. Xie [2] formulate the exam timetabling problem as a graph coloring problem. In the paper, the initial solution is generated from an algorithm derived from a bin packing algorithm (i.e. largest enrolment first). The initial solution might be feasible or might not be. New solution is always generated from an atomic move where only exactly one exam x in s (current solution) is moved from a timeslot T_i to another timeslot T_j . The move is denoted as (x, i, j) . The size of s' , the neighbours of s depends on the size of candidate exams and the number of timeslots.

L. Paquete & T. Stutzle [3], adapts tabu search technique in their work to a lexicographic optimization of the

timetabling problem. The timetabling problem is first attacked by order of all the constraints which exists in the problem domain in a hierarchy according to their importance. L. Paquete and T. Stutzle (2002) conclude from their research that the length of the tabu list must increase when more constraints are added into the objective function.

4 DESIGN

4.1 Representation of the problem

We use an undirected graph $G = G(V,E)$ to represent the exam timetabling solution. The following notations are used to represent the exams, students taking exams and timeslots allocated for exams.

Let

V – The exams (or set of node in graph), i.e. $V = (v_1, v_2, \dots, v_i)$ where i is the i^{th} exam

s_i – The number of students taking exam v_i

T – A set of consecutive timeslots, each contains zero or more exams i.e. $T = (T_1, T_2, \dots, T_k)$, where k is the number of timeslots.

E – The edge sets in the graph, e_{ij} edge between node v_i and node v_j if there is at least one student who is taking exam v_i and v_j

m – Weight of edge e_{ij} on the graph.

e – Total number of edge in the graph

W – The sum of the weights of set of edges E .

For an exam timetable to be feasible, the sum of edges having both endpoints in the same timeslot must be zero. In addition, the sum of weight of edges between the timeslots with a given distance (i.e. of adjacent timeslots) should be minimized in order to produce an optimal timetable. For any soft constraint violated, penalties will be assigned. Soft constraints are categorized and prioritized. Penalties increases as the constraint order increases.

4.2 The Objective function

Objective function is the most important component to drive the search process. We define our objective function as below :

Let

$E_{i,0}$ – set of edges having both endpoints in T_i ($i = 1,2, \dots, k$), considering first order conflict;

$E_{i,1}$ – set of edges having both endpoints between T_i and T_{i+1} ($i = 1, 2, \dots, k$) considering second order conflict ;
 $E_{i,2}$ – set of edges having both endpoints between T_{i+1} and T_{i+2} ($i = 1, 2, \dots, k$) considering third order conflict ;
 W_0 – the sum of the edges set $E_0 = \cup E_{i,0}$ ($i = 1, 2, \dots, k$)
 W_1 – the sum of the edges set $E_1 = \cup E_{i,1}$ ($i = 1, 2, \dots, k$)
 W_2 – the sum of the edges set $E_2 = \cup E_{i,2}$ ($i = 1, 2, \dots, k$)

The objective function is define as

$$f(s) = p_0 \times W_0 + p_1 \times W_1 + p_2 \times W_2$$

where

s – exams condition for all T_i
 p_0, p_1, p_2 – Penalties for different weight constraints in s that are characterized by W_0, W_1, W_2

For the problem that we are going to solve, $p_0 \gg p_1 \gg p_2 = 1$ where p_0 is penalties for first-order conflict, p_1 for second-order conflict and p_2 for third-order conflict. Hence, the problem of optimizing the exam timetable is equivalent to minimizing the value of the objective function $f(s)$ of a solution set S .

4.3 The Penalty weighting

The soft constraints are embodied into the objective function f to measure the quality of the solution. Violation of constraints will increase the value of the objective function. In order to produce good quality exam timetabling solutions, exams on the same day or more spacing between exams taken by a same student are the main concern when designing the penalty scheme. Consecutive timeslots and days of exam should at least lay one unit apart.

4.4 Neighbourhood structure and move selection

We define the solution as having m exams scheduled at n timeslots. Let s be the solution, for each iteration a new solution s' is obtained by moving an exam to a new timeslot. The neighborhood $N(s)$ can then be defined as an assignment of any exam m to any timeslot n .

Two types of candidate list denoted by V and V^* are used to form the neighborhood $N(s)$ of the search space from current solution. The V candidate list will contain all the exams (nodes in the graph) and the V^* candidate list contains only those exams (nodes) involved in at least one conflict in the current solution. Alternatively, different type of candidate is used to generate the neighbourhood, beginning with the V^* type candidate. The size of neighbourhood $N(s)$ depends on the size of candidate list and the number of timeslots.

4.5 Candidate list strategy

In each iteration, the neighbourhood regions consist of all the admissible moves. We then enforce a kind of tournament selection over all the admissible moves; moves with minimum objective costs will be selected. For the exam timetabling problem, there might be more than one move that give the minimum objective value, a candidate list is used to keep all the promising moves. In each iteration, every admissible move will be compared among each other. If move (x, i, j) gives objective cost better than the current cost, we clear the candidate list and start the candidate list with the move (x, i, j) , else we keep the move (x, i, j) as a new node at the beginning of the candidate list.

4.6 Tabu Relaxation

Let r be the maximum tabu relaxation iteration. At any given number of r iteration if the long term tabu list is full since the last best solution was found, or current solution is worse then the last best solution, tabu relaxation happens where all the entries in long term tabu list and short term tabu list will be emptied before the start of the next search iteration. The purpose of tabu relaxation is to drive the search into new region and increase the likelihood of finding a better solution.

4.7 Short-term tabu list

We implement two types of tabu lists namely *ts-value* and *ts-variable*. *ts-value* is used to keep a move attribute for both the changes of timeslot and the exam as tabu. *ts-variable* only records the exam that being moved as tabu. With *ts-variable*, we forbid the selection of exam E_i to be moved to any other timeslot T_i . With *ts-value*, the recent moves of exam E_i to T_i will not be selected in the subsequence iteration to form the neighbourhood region.

4.7.1 Long-term tabu list

At the beginning of search iteration, the objective cost decreases in great percentage from one iteration to another. As searching iteration increase, the improvement of objective cost will get smaller and smaller. This is due to the good moves that make the objective cost decrease in great percentage are already being keep inside the short term tabu list. When there are no changes or little changes of objective cost within some continuous iterations, cycling might happen within the short term tabu list. Hence diversification is required to drive the search to another space where better solution might be found. For our timetabling problem, we used frequency-base memory structure to diversify the search.

5 RESULT

We conducted experiments to investigate the effects of the different tabu list over the quality of solution produced. Tabu list of different length with two different maximum null iteration and maximum tabu relaxation values were

tested. Actual data from Universiti Sains Malaysia for semester 1 of academic year 2001/02 were used. Table 1 shows the properties of the data set used in the experiments.

Table 1: Properties of Dataset

Properties of Data Set	Data set S01021
Number of exams	574
Total Timeslots	40
Constraints :	
Forbidden time slots	1
Groups of concurrent exams	29
Pair of conflicting exams	9550

5.1 Using only short term tabu list

This experiment was carried out without the long term tabu list. The short term tabu list was varied from 5 to 15, while two different maximum null iteration and maximum tabu relaxation combinations, i.e. 600/400 and 300/200 were tested. The results are presented in figure 1.

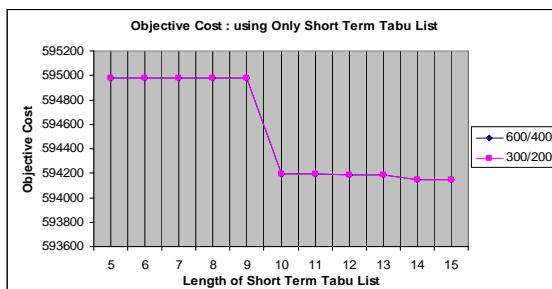


Figure 1: Short term tabu list only

When the long term tabu list is disabled, the objective cost under two different maximum null iteration and maximum tabu relaxation (i.e. 600/400 and 300/200) are the same for this data set.

Short term tabu list alone does not contribute significantly for the search in this particular experiment. Although the longer short term tabu list performed slightly better than the shorter short term list.

5.2 Using only long term tabu list

This experiment was carried out without the short term tabu list. The long term tabu list was varied from 50 to 550, while two different maximum null iteration and maximum tabu relaxation combinations, i.e. 600/400 and 300/200 were used. The results are presented in figure 2.

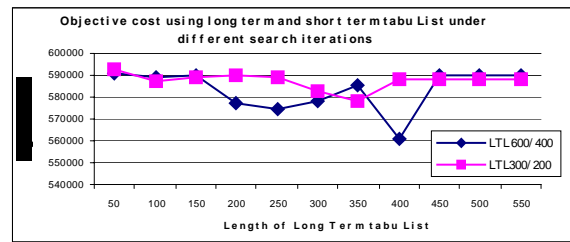


Figure 2: Long term tabu list only

When shorter length are used for long term tabu list (i.e. from 50 to 150), objective cost differences between two different search iterations are small. However, when the length of the long term tabu list is set between 200 and 450, the objective cost improved by some percentage for the higher search iteration. This suggests that for long term tabu list, when the length is set within approximately between 20% and 80% of the total number of exams that need to be scheduled, better cost can be found with longer search iteration.

The results obtained using only long term tabu list compared to using only short term tabu list shows that long term tabu list helps to improve the quality of solution in much more higher percentage. The setting of the length of long term tabu list does effect the results obtained.

5.3 Using both long term and short term tabu list

To learn the effect of using both, the long term and short term tabu list together in guiding the search process, we tested using different combination of short term and long term tabu list parameter as we did for the previous experiments. Two different maximum null iteration and maximum tabu relaxation combinations, i.e. 600/400 and 300/200 were used. The results for 600/400 combination is presented in figure 3 while the results for 300/200 combination is presented in figure 4.

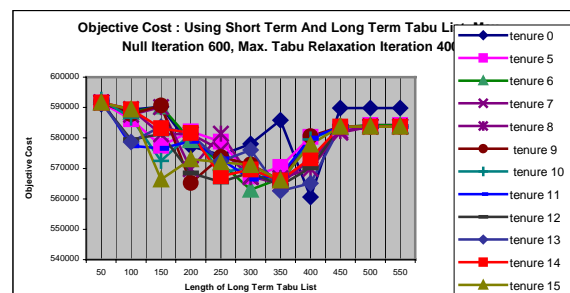


Figure 3: Short and long tabu list with 600/400

Best results are obtained for longest short term tabu list (i.e. length 14 and 15) with medium size long term tabu list (i.e. length between 300 and 400). Good results were also obtained for shorter short term tabu list with medium size long term tabu list. The results were much worst for shorter long term tabu list (i.e. length between 50 to 250).

and longer long term tabu list (i.e. length between 450 and 550).

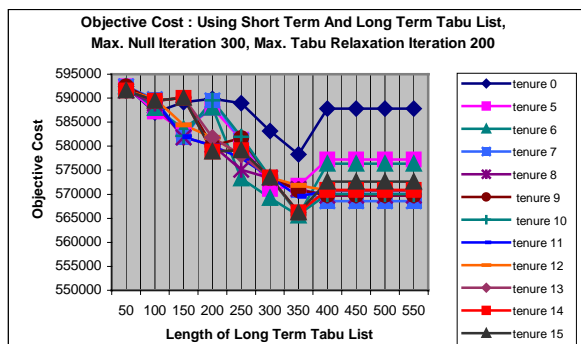


Figure 4: Short and long tabu list with 300/200

Best results were obtained when the long term tabu list was set at 350. Short long term tabu list produced the worst, but the results improved linearly until it peaked around 350. Thereon the results deteriorated slightly. Generally this is true for short term tabu list of medium to long size.

6 CONCLUSION

The intelligent inside tabu search comes from the adaptive memory and responsive exploration. The long term and short term memory play very important role in improving the efficiency of exploration process in the neighbourhood search. These memories structure are used to keep track of the information related to the searching process.

We use different parameters for long term tabu list and short term tabu list to find the formation that yields good timetabling solution. Based on the results obtained using the dataset, we compared and among our findings includes the following:

- Using short term tabu list alone can't make much improvement though different search iterations were used.
- Compared to short term tabu, long term tabu list when used, give better improvement in term of the quality of solution.
- Compared to using only single type of tabu list, combined use of both the long term and short term tabu list produced better results.
- The length used for long term tabu list is critical to the success in finding better solution. Too short the length, little or not improvement can be found. Too long the length, the quality of solution decrease by small percentage. With the length set around 2/3 of the number of total exams that we need to schedule, better solution can be found.

7 REFERENCE

- [1] A.Scharef and L. Di Gaspero, *Tabu Search techniques for examination timetabling*. In E.Burke and W. Erben, editors, Proc of the 3rd Int. Conf. On the practice and Theory of Automated Timetabling PATAT'02, number 2079 in Lecture Notes in Computer Science, pages 104 – 117. Springer-Verlag, Berlin-Heidelberg (2001)
- [2] George M. White and Bill S. Xie. *Examination Timetables and Tabu Search with Longer Term memory*. School of Information Technology and Engineering, University of Ottawa. Ottawa, K1N 6N5, Canada. (2000)
- [3] L. Paquete & T. Stützle. Empirical Analysis of Tabu Search for the Lexicographic Optimization of the Examination Timetabling Problem. Accepted to the *Practice and Theory in Automated Timetabling Conference (PATAT 2002)*. Also available as Technical Report AIDA-02-02, Fachgebiet Intellektik, Fachbereich, Informatik, Technische Universität Darmstadt (2002)
- [4] Michel Gendreau, *An Introduction to Tabu Search*, Centre de recherche sur les transports and Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, Canada (2002)