

Tabu Analysis List Length on Lecturing Scheduling

Rayuwati¹, Husna Gemasih¹ and Mursalin²
{ rayuwati@ugptakengon.ac.id }

¹Informatics Program Study, Universitas Gajah Putih, Aceh, Indonesia

²Department of Mathematics Education, Universitas Malikussaleh, Aceh Utara, Indonesia

Abstract. Lecturing scheduling is the management of lecture plans that will be arranged in period and room and lecturer teaching lectures. The lecturing scheduling process is well arranged by fulfilling a set number of constraints with the goal of producing a more optimal lecture schedule. Taboo algorithm searches a search-based optimization method by shifting from one solution to another by selecting the best solution from the current (current) environment that is not classified as a taboo solution. The fundamental memory structure in taboo search is called taboo list. The taboo list stores the attributes of some moves (the solution transitions) in the previous iterations. Tabu search algorithm is influenced by several parameters, one of which is the magnitude of the taboo list will affect the iteration and time provided to produce a more optimal lecture schedule.

Keywords: Scheduling, Lectures, Tabu search, Tabu list

1 Introduction

In the academic system of college, scheduling is a routine job done every semester. Scheduling is used to apply events that contain the components of the course, lecturer, class and semester on the time slot that contains the time and space components as needed. If using a manual system then this problem takes a long process time to look for the solution, especially when problems arise with the increase in the number of components and the constants or conditions specified by the institution where the schedule is used [1].

In this case constraints must be fulfilled have a number of specialties due to limited time and space and the willingness of lecturers in teaching with the time specified, because some lecturers have activities outside the campus. Allocating lecturers, time, and space to a class will greatly affect other classes to be a difficult case to solve. Problems are still increasing complex with the course that must be allocated in a certain room that has its own criteria.

Tabu Search is a search-based optimization method through a shift from one alternative to the next, selecting the best alternative Neighbourhood now (current) that is not considered a taboo alternative. The basic idea of a taboo search algorithm is to prevent searching from local searches from retrieving the already traced solution space by utilizing a memory structure that records some of the traces of the search process that has been performed. Based on the description above, the authors are interested to take the title "Tabu Analysis List length On lecture scheduling"[2].

1.1 Problem Formulation

Based on the background for the problems that have been raised, then the formulation to the problem is the analysis of the effect from the tabu length list on tabu.

2 Literature Review

2.1 Tabu Search

According to S. Abdullah, Ahmadi, Burke, Dror, & McCollum [4] tabu search is an optimization method based on local search. The search process moves from one solution to the next, choosing the best alternative from the current alternative environment (taboo) that is not classified as a forbidden alternative (taboo). The fundamental memory structure in taboo search is called taboo list. The taboo list stores the attributes of some moves (the solution transitions) in the previous iterations.

2.2 Algoritma Tabu Search

According to Glover (1989) as a guide to move from i to the next solution in $N(i)$. memory usage as a barrier in the selection of several subsets of $N(i)$ by preventing movement to some neighboring solutions.

- Selects initial solution I in S . Set $i^* = I$ and $k = 0$.
- Set the value $k = k + 1$ and generate subset V^* as the solution in $N(i, k)$
- Select j 'best' in V^* and set $i = j$.
- If $f(i) < f(i^*)$ then set $i^* = i$.
- If a stop condition is found, then the process is terminated, whereas if it has not returned to step 2.

2.3 Research Contribution

The author hopes this research can be use dask now ledge and compareson insight about scheduling and search tab algorithm and can be applied to real life such as the problem of arranging schedule, job shop scheduling, map coloring and storage area.

3 Method

Problems discussed in this research is to analyze the size of taboo list on lecturing scheduling process with taboo search algorithm. The analysis is done by comparing the number of steps and time required by the taboo search algorithm to generate more optimal lecture scheduling with different taboo list sizes. The research methodology in this study is illustrated as in Figure 1.

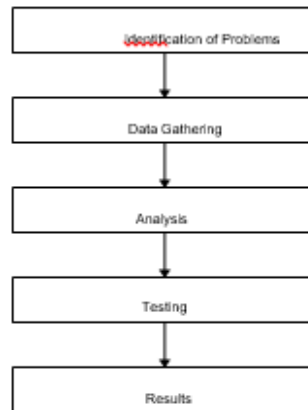


Figure 1: Research Method

3.1 Analysis

The scheduling of the lecture is a system of managing the lecture process at a university in the form of a set of course C ($c_1, c_2, c_3, \dots, c_n$) that will be arranged in a set of periods T ($t_1, t_2, t_3, \dots, t_n$) room R ($r_1, r_2, r_3, \dots, r_n$). Each C_i lecture program consists of several lectures that will be scheduled. In this study, the scheduling process discussed consists of several entities, among others.

- Day, TimeSlot, and Period: Scheduling of days used in 1 week of lectures and some equal timeslot for entire lecture day. A period is a pair of days as well as the available timeslot
- Lectures and Lecturers Program: A lecture program is a collection of lectures and lecturers who teach in the lectures and students who take the lecture.
- Room: space for lecture program implementation. Each room is considered appropriate for every lecture
- Curriculum: is a set of courses that will be selected by all students.

The scheduling rules used in this study are divided into 2 parts: hardconstrain and softconstrain. Hardconstrain deals with:

- Lectures: Each lecture should be scheduled for different periods and times.
- Room: class schedules should not be arranged in the same room and period
- Conflict: Lecture programs in the same curriculum or taught the same teacher can not be scheduled in the same period, or in other words each period does not have the same student or the same teacher.
- Availability: If the curriculum teacher is not available within the given period, the course schedule is also not available.
- If the curriculum teacher is not available within the given period, then the course schedule is not available.

While softconstrain is a rule of penalty (penalty) against the scheduling conducted in the form:

- Space Capacity: students who are scheduled in the curriculum of a room according to the given space capacity

- Room Stability: All lectures are arranged in the same room. If it can not be done then the amount of space used to a minimum
- Minimum Working Days: Each lecture is endeavored to be disseminated by minimizing the day
- Curriculum Density: Student lecture agenda cultivated to a minimum

3.2 Data Format

Data is made in the order that has been determined that Courses Courses, Room, Curriculum, and Constrain. The data format performed in the scheduling is arranged in the form of:

- Program: <ProgramID><Lecturer>
<#student><MinWorkingDays>
<# lecture>
- Room: <RoomID><capacities>
- Kurikulum: <KurikulumID><# Program><lectureID-1> ... <lecture-n>
- Constraints: <ProgramID><Day><Periode>

as an example :

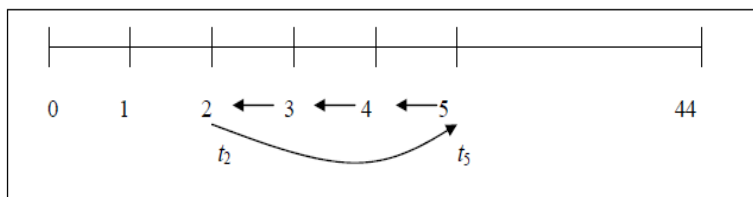


Figure 2: Proses Swap Move [2]

Program : <Program1><Dsn.X><2><3><30>
Room : <Room1><35>
Curriculum : <Curriculum1><Program1>
Constraints : <Program1><2><0>

The data tested in this research are data: 24 lecturers, 14 curriculum, 30 lectures, 160 lectures, 30 periods, 6 rooms and 53 periods are available to fill.

3.3 Tabu Search

Tabu search uses tabu-lists to deny solutions that meet certain attributes to prevent search processes from experiencing cycling in the same solution area, and guide the search process through unvisited solutions [5].

According to [6] Muller, Camozzato, & Araujo [5] in the design of lecturing scheduling using tabu search, the initial schedule that has been produced serve as the initial solution. The next process is to look for conflicts that occur during each schedule period on each iteration. The total value of the conflicts obtained is used as the best value temporarily. So get the initial solution stored in the taboo list for the next iteration process.

Calculation of cost in Tabu Search algorithm in this thesis is divided into 2, the value of Hard Constrain cost and Soft Constrain cost value. Value of Hard Constrain cost. The cost

value of Hard Constrain has been designed to produce a value of 0, if the schedule already has the best lecturers' composition and lecture time, fulfill all the constraints.

Instead the SoftConstrian cost is designed to approach 1. If SoftConstrain value close to 1 means that the resulting schedule contains many violations.

3.4 Neighborhood

According to Betrianis & Teguh [6] neighborhood is a solution found from the issues being discussed. Neighborhood is found through the moving stages of each item in the Taboo Search algorithm. In this study the selected neighborhood is N2 (select a random schedule and move to a new randomly scheduled place as well), where the solution is found through a schedule exchange with no violation of the constraint set. SwapMove process and examples of scheduling swap process in the picture below.

Monday	Tuesday	Wednesday	Thursday	Friday
1	9	17	25	33
2	10	18	26	34
3	11	19	27 lesson i'	35
4	12	20	28 lesson i'	36
5 lesson i	13	21	29	37
6 lesson i	14	22	30	38
7	15	23	31	39
8	16	24	32	40

Figure 3. Swap Move Proses schedule

3.5 Tabu List

According to Abdullah, Burke, & Mccollum [1] tabu list is to store n previously visited solutions. So any current solution left behind will be recorded and entered into a list called tabu list. Taboo list is also to avoid the acceptance of previous solutions into current solutions in the next loop.

Before a neighbourhood solution is checked, it must first be checked whether the solution has been entered into the taboo list. If it is then the neighbour solution should be rejected and the process is continued by checking the new neighbors solution. Thus with the taboo list can avoid cycles (iteration goes on without improving the quality of the solution).

In order for the desired solution to be achieved, each soft constraint violation is defined as a different cost value. This is done because the level of necessity to fulfill a soft constraint is different from one to another. Soft constraints that need to be filled are given a higher weight than soft constraints that are considered more tolerable.

The problem of tabu list size in the course scheduling process is done to determine the schedule that is considered tabu (meet the solution) at the time the program is running. Schedule that is considered to meet is placed into a tabu list to avoid repeated searches.

The tabu list process is used to store tabu data. If the taboo list size is 2, then every time the tabu data is stored as much as 2. If the tabu schedule data is more than 2 then the first tabu data will be removed from the tabu list because the process on the tabu list is FIFO (First In First Out). If tabu list 2 [3].

In this study, the taboo list value to be tested consists of several taboo list sizes, among others 2, 4, 6, 8 and 10. The scheduling test in this study with various taboo list sizes was

performed to determine the number of iterations and time required in order to produce a more optimal schedule.

4 Results

The results obtained in this research is the output of processing schedule tabu search algorithm with taboo list size is 2, 4, 6 and 8. Testing system in this research is done on laptop with specification of i3 core processor and 2 GB memory. The discussion of this thesis is done on the time and number of iterations obtained for each taboo list size in the roster scheduling process.

4.1 Parameter

The parameters tested in this study include the taboo list size, the number of iterations, and the time required to achieve a more optimal schedule.

4.1.1 Program Output

The output generated in this study is the output of the program, as well as the results of the tests performed on the parameters set, which is the initial schedule compiled starting on Monday, Tuesday, Wednesday, Thursday and Friday. Test results on lecture schedules with taboo list size of 2, 4, 6, 8, 10 and description of test results against the taboo list above by giving HardConstrtain and SoftConstrtain values and the average time required to achieve an optimal schedule.

Day	Time	B	C	E	F	G	S	Unassigned
0 Mo	0	c0001.2	c0002.7	c0072.3	c0061.3	c0063.1	c0057.3	
	1	c0001.1	c0003.2	c0072.2	c0062.0	c0063.4	c0057.2	
	2	c0016.6	c0025.6	c0068.1	c0062.3	c0059.4	c0064.4	
	3	c0015.0	c0024.3	c0068.0	c0059.1	c0071.3		
	4	c0009.5	c0015.6	c0069.2	c0030.1	c0071.2		
1 Tu	0	c0002.4	c0015.3	c0076.5	c0070.0			c0058.1
	1	c0001.0	c0015.5	c0069.2	c0061.0			c0064.2
	2	c0015.2	c0025.7	c0069.1	c0005.0			c0064.1
	3	c0016.5	c0024.2	c0076.2		c0057.0		c0057.2
	4	c0016.0	c0002.0	c0076.1				
2 We	0	c0002.2	c0072.2	c0059.3	c0011.1			
	1	c0002.1	c0072.1					
	2	c0001.4	c0011.4	c0076.5	c0061.0	c0059.3	c0064.3	
	3	c0004.1	c0033.0	c0072.6	c0061.5	c0063.0	c0067.4	
	4	c0005.1	c0024.1	c0058.4	c0062.2	c0063.3		
3 Th	0	c0005.2	c0024.0		c0030.3	c0071.1	c0066.2	
	1	c0005.2	c0024.0		c0030.3	c0071.1	c0066.2	
	2	c0002.3	c0072.3		c0014.0			
	3	c0004.0	c0033.0	c0069.5	c0061.2	c0066.3	c0065.1	
	4	c0004.5	c0033.3	c0068.4	c0061.1	c0066.4	c0065.5	
4 Fr	0	c0004.6	c0033.2	c0068.5	c0059.0	c0066.5	c0076.0	
	1	c0001.5	c0015.7	c0069.0	c0070.4	c0066.2	c0058.6	
	2	c0001.3	c0015.4	c0064.4		c0066.1	c0058.2	
	3	c0002.5	c0015.1	c0067.6	c0011.1	c0066.0	c0058.0	
	4	c0004.4	c0025.6	c0072.6	c0061.4	c0063.2	c0064.4	

Figure 4. Initial Schedule View

Figure 4 is an initial schedule that is compiled and used as an initial solution. The composed schedule pairs are placed in days 0 (Monday), 1 (Tuesday), 2 (Wednesday), 3 (Thursday), and 4 (Friday).

Test results on lecture schedule with taboo list size of 2.

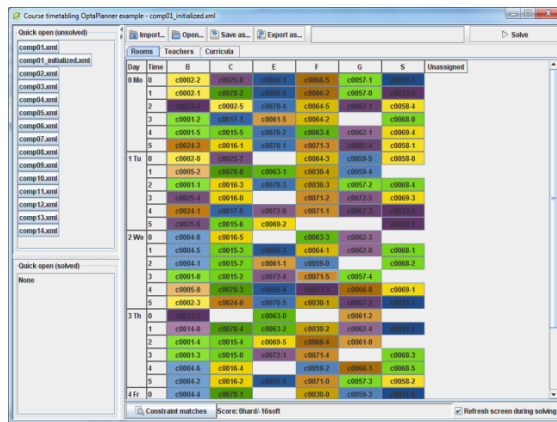


Figure 5. Test Results with Taboo Size List 2

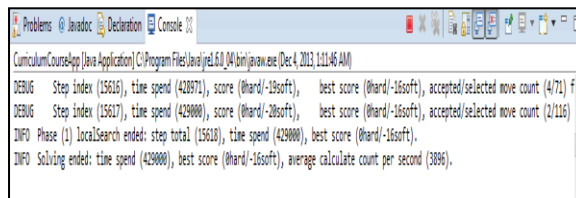


Figure 6: Description of Testing Results with Tabu List 2

Test results on lecture schedule with tabu list size of 4.

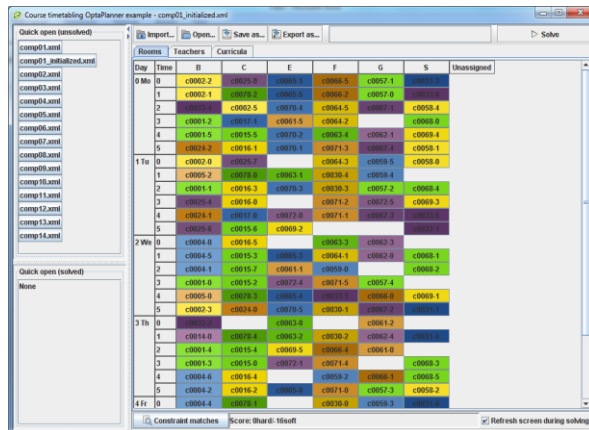


Figure 7. Test Results With Tabu Size List 4

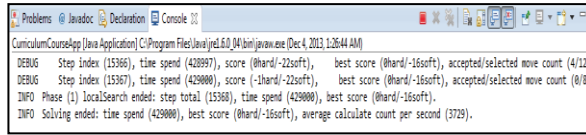


Figure 8: Description of Testing Process Results with Tabu List 4

Test results on lecture schedule with tabu list size of 6

Room	Teachers	Curricula
0	B	C
1	B	E
2	B	F
3	B	G
4	B	S
Unassigned		

Figure 9: Test Results With Tabu Size List 6

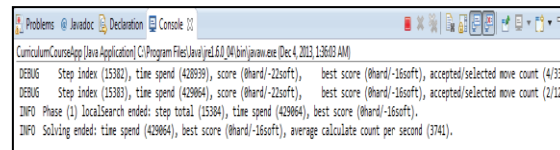


Figure 10: Description of Testing Process Results with Tabu List 6

Test results on lecture schedule with tabu list size of 8 .

Room	Teachers	Curricula
0	B	C
1	B	E
2	B	F
3	B	G
4	B	S
Unassigned		

Figure 11. Test Results With Tabu Size List 8


```

Problems | Javadoc | Declaration | Console
CurriculumCourseApp [Java Application] C:\Program Files\Java\jdk.6.0_04\bin\javaw.exe (Dec 4, 2013 2:24:43 AM)
DEBUG Step index (15483), time spend (420995), score (0hard/-26soft), best score (0hard/-16soft), accepted/selected move count (4/488)
DEBUG Step index (15483), time spend (420900), score (0hard/-41soft), best score (0hard/-16soft), accepted/selected move count (8/19)
INFO Phase (1) localSearch ended: step total (15484), time spend (420900), best score (0hard/-16soft).
INFO Solving ended: time spend (420900), best score (0hard/-16soft), average calculate count per second (3794).

```

Figure 12. Description of Testing Process Results with Tabu List 8

Test results on lecture schedule with taboo list size of 10

Figure 13. The Results With Tabu Size List 10

```

Problems | Javadoc | Declaration | Console
CurriculumCourseApp [Java Application] C:\Program Files\Java\jdk.6.0_04\bin\javaw.exe (Dec 4, 2013 2:26:18 AM)
DEBUG Step index (15374), time spend (420894), score (0hard/-22soft), best score (0hard/-16soft), accepted/selected move count (4/374)
DEBUG Step index (15375), time spend (420900), score (0hard/-26soft), best score (0hard/-16soft), accepted/selected move count (8/24)
INFO Phase (1) localSearch ended: step total (15376), time spend (420900), best score (0hard/-16soft).
INFO Solving ended: time spend (420900), best score (0hard/-16soft), average calculate count per second (3735).

```

Figure 14: Description of Testing Process Results with Tabu List 10

4.2 Discussion

Results of experiments conducted several times, then obtained the results of taboo tabbed list size analysis taboo search algorithm on lecture scheduling process in the table below

Table 1. Tabu Size List Analysis Result

No	Tabu List	Number of Iterations	Time (Mili Second)
1	2	10612	429005
2	4	10301	429001
3	6	9524	429001
4	8	9234	429001
5	10	9234	429001

Base on the experimental results in table 1. It is found that the larger the taboo list size will reduce of iterations required by the taboo search algorithm to reach the optimum schedule.

Results of Hard Constrain value obtained from the resulting schedule is 0, which gives meaning that there sulting schedule meets the specified criteria. There sulting Soft-Constrain value is-16.

5 Conclusion

The conclusions that can be produced in this study include:

- The taboo list size will affect the taboo search algorithm in the lecturing scheduling process. The larger the taboo list size, the less iteration it will take optimal lecture schedule.
- Average time required the optimal schedule is 429001.6 mili seconds.

References

- [1] S. Abdullah and K. Shaker, "Construction of course timetables based on great deluge and tabu search," *Proc. MIC ...*, no. May 2014, 2009.
- [2] S. Abdullah, "Heuristic Approaches for University Timetabling Problems," no. June, p. 226, 2006.
- [3] 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," *J. Oper. Res. Soc.*, vol. 58, no. 11, pp. 1494–1502, 2007.
- [4] F. Glover, "Tabu Search - Part I," *ORSA J. Comput.*, vol. 2 1, no. 3, pp. 4–32, 1989.
- [5] F. Glover, "Tabu Search - Part II," *ORSA Journal on Computing*, vol. 2, no. 1. pp. 4–32, 1990.
- [6] F. Muller, M. Camozzato, and O. Araujo, "Neighbourhood constraint for a Tabu search algorithm to schedule jobs on unrelated parallel machines Neighbourhood Constraint for a Tabu Search," no. June 2014, 2001.
- [7] Betrianis and P. Teguh, "Penerapan Algoritma Tabu Search Dalam Penjadwalan Job Shop," vol. 7, no. 3, pp. 107–112, 2003.
- [8] S. Abdullah, E. K. Burke, and B. Mccollum, "Using a randomised iterative improvement algorithm with composite neighbourhood structures for the university course timetabling problem," *Metaheuristics*, pp. 153–169, 2007.