International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2278-3075, Volume-7 Issue-5S, January 2019

Academic Timetable Optimization For Asia Pacific University, Malaysia using Graph Coloring Algorithm

Sireesha Prathi Gadapa, M. M. Mohamed Ubaidullah, Rajasegeran Ramasamy, Raja Rajeswari Ponnusamy

Abstract: ***This paper proposes an optimized solution for timetable scheduling system (TTSS) at Asia Pacific University[APU], Malaysia. TTSS generates the students' weekly timetable under given set of constraints and preferences such as lecturers, classroom availability, modules etc. Due to various coinciding problems (multiple intakes, various cohorts, academic calendar, increase in student population and limitations of resources), TTSS optimization is vital. This optimization of TTSS is known as Node-Point problem, for which there is no known polynomial algorithm, leads to time consuming problem to find an optimizesolution that grows rapidly with population size. Graph coloring algorithm is a heuristic approach to generate an automated system that will optimize the TTSS to provide a conducive learning environment for students and utilize absolute APU's resources in an efficient way. The paper propose a user-friendly database system that applies graph vertex coloring approach associating with a data course matrix to overcome the problems of TTSS, APU.***

Keywords: Graph coloring, Timetable scheduling system, Heuristic approach, Course matrix

I. INTRODUCTION

Timetable scheduling system [TTSS] is a typical and vital problem faced by Asia Pacific University [A.P.U], Malaysia due to the multiple intakes, various cohorts, realistic constraints and preferences. A.P.U offers many programs and each program has many modules. Therefore, scheduling students' weekly timetable become more complex task especially when scheduling the Lecturers to the modules and the students to individual classes without having any conflict. The problem of generating clash-free timetable is as hard as solving as NP-complete problem. The realistic constraints are divided into hard and soft constraints. When creating scheduling the modules and common modules of each program during a semester, the following are the typical hard constraints (Vinod J. Kadam, Samir S., 2016).

Constraint 1: Assigning Lecturers, students of various cohorts, classrooms and timeslots without clashes.

Constraint 2: Meeting the required number of contact hours associated to the curriculum of the modules of each program and the required number of modules and common modules during a semester.

Constraint 3: The total allocations required must not be greater than the resources available.

Many different techniques have been applied to scheduling timetable problems. This paper is concerned with the problem of course timetable scheduling, where the coloring can provide an algorithm which will generate the timetable with no conflicts. Thus, an optimal solutions to the problems concerned can be found by finding minimal colorings for the corresponding graphs. Both decision and optimization versions are precisely defined using Graph coloring algorithm. This can benefit APU through efficient scheduling.Graph Coloring algorithm is a one-to-one mapping of vertices to colors such that adjacent vertices are assigned to different colors. Graph coloring problem (GCP) consists in finding minimum k-color existence also called chromatic number denoted by χ. GCP is a NP-hard problem and one of its application is timetable scheduling (Ayanegui and Chavez-Aragonar, 2009).

Many studies on Timetable scheduling system (TTSS) problem has been conducted around the world where various algorithm has been suggested to provide an optimized solution to the problem. Among the suggested methods are cluster algorithms, sequential, constraints based and meta-heuristics. Cluster algorithms by Desroches is one of the earliest methods used to solve the timetabling problem (Assi, 2018). Sequential algorithms which seem to be more effective in obtaining optimize solution compared to cluster were introduce in late 80s. Studies conducted on the effectiveness of this methods found out the sequential method can be improved by blended with various approaches (Wera, 1985). Graph coloring method is one of the hybridized methodology of the sequential algorithm. Graph coloring algorithms is being used widely in finding feasible or optimize solution towards solving timetabling problem (Assi, 2018). The rationale of using graph coloring is to help to identified conflicted arise while allocating process in scheduling the timetable. Border, 1964 is one of the pioneer in working with graph coloring algorithm to obtain feasible solution to this problem. However, the approaches taken here is more to avoid or minimize the conflicts rather than eliminate them. The method was improved by introduction of matrices in the process. Woods and Matula, Marble and Isaacon blend matrices in the graph coloring approach.

Revised Manuscript Received on January 19, 2019.

Sireesha Prathi Gadapa, School of Mathematics, Accounting and Quantitative Studies, Asia Pacific University, Malaysia.
M. M. Mohamed Ubaidullah, School of Mathematics, Accounting and Quantitative Studies, Asia Pacific University, Malaysia
Rajasegeran Ramasamy, School of Mathematics, Accounting and Quantitative Studies, Asia Pacific University, Malaysia.
Raja Rajeswari Ponnusamy, School of Mathematics, Accounting and Quantitative Studies, Asia Pacific University, Malaysia.

More studies were conducted as the result from the tested method did not satisfied the need of an optimize solution to the problem as various improvement methods emerged and able to generate more feasible solution. Anyhow, graph coloring achieve the objective of identifying the conflict in scheduling timetable at early stage.

A. Problem statement

APU is providing undergraduate programs such that Business, Accounting and Finance which require five modules for each program during a semester. In order to schedule the modules to the students, schedulers need to identify the availability of module lectures. In this section, we have presented 2 distinctive cases of scheduling problem and their conflict free solution timetables have been presented.

B. Scheduling Problem

The first scheduling problem involves Business, Accounting and Finance programsmodule combination. The problem arise due to students from different programs are taking up modules across the program which leads to complexity of scheduling. Thus, Table 1 shows the modules taken by student in semester 1, level 1 in April 2018 intake for different programs and specializations. Based on Table 1, there are 4 different program combinations that students can register in Semester 1, Level 1 for Business, Accounting and Finance programs and 9 unique modules (*)have been chosen in April-2018intake for this study.

Program	Specialization	Code	Module Name
Business	Business Management	IMT	Introduction to Management (*)
	Business Management with E-Business	BCS	Business Communication Skills(*)
	International Marketing Management	CITW	Computing and IT in the workplace(*)
	Human Resource Management	FEP	Fundamental of Entrepreneurship (*)
		QSS	Quantitative Skills (*)
Business	Tourism Management	IMT	Introduction to Management
		BCS	Business Communication Skills
		CITW	Computing and IT in the workplace
		QSS	Quantitative Skills (*)
		ISMT	Introduction to Service Management (*)
Accounting & Finance	Accounting & Finance	IMT	Introduction to Management
	Accounting & Finance With Forensic	BCS	Business Communication Skills
	Accounting & Finance With Taxation	FEP	Fundamental of Entrepreneurship
	Accounting & Finance with Forex and Investments	QSM	Quantitative and Statistical Methods (*)
		FA	Financial Accounting (*)
Accounting & Finance	Accounting & Finance With Internal Audit	IMT	Introduction to Management
		BCS	Business Communication Skills
		QSS	Quantitative Skills
		QA	Quantitative Analysis
		FA	Financial Accounting

Table 1: List of programs and combination of modules

C. Hard Constraints:

- Assigning Lecturers, students of various cohorts, classrooms and timeslots without clashes.
- Meeting the required number of contact hours associated to the curriculum of the modules of each program and the required number of modules and common modules during a semester.
- The total allocation required must not be greater than the resources available.

D. Soft Constraints:

- Concerning the limits on the workload of Lecturers in A.P.U. setting.
- Scheduling not more than two lecture classes for a lecturer in a day.

II. METHOD & MATERIAL

This method is designed to cater as a solution to the above problem. The following graph coloring algorithm illustrate the methodology.

Input data set: Combination of modules for different programs with specialization.

Output data set: The minimum number of conflict free time slots required to schedule modules.

A. Graph Coloring Algorithm:

Step 1: Identify the list of programs and combination of courses
Step 2: Generate adjacency matrix based on combination of courses
Step 3: Input the conflict graph G.
Step 4: Calculate the degree sequence of each module for the conflict graph G.
Step 5: Assign color A to the vertex of G having highest degree.
Step 6: Assign color A to all non-adjacent uncolored vertices of v and store into Colored_Array.
Step 7: Assign new color, color B which is not previously used to the next non- colored vertex having highest degree.
Step 8: Assign color B to all non-adjacent uncolored vertices of v and store into Colored_Array.
Step 9: Repeat step 6 – 7 until all vertices are appropriately colored.
Step 10: Set total number of colors used in Colored_ array is equal to minimum number of non-conflicting timeslots.
Step 11: End

B. Module-Lecturer Problem

Problem statement 2

In Asia Pacific University there are m lecturers l_1, l_2, …. l_m and n modules m_1, m_2, …, m_n to be taught. Given that lecturer l_i is required to teach module m_j for p_{ij} timeslots.For a given ‘n’ number of modules, ‘m’ number of lectures and there are ‘p’ number of timeslots available and a timetable needs to be scheduled.

Hard Constraints:

- Scheduling the modules and classrooms to the lecturers and students should be allotted at a fixed timeslot.
- Each module can be handled by one and only lecturer in a semester.

Soft Constraints:

- Lecturer taking more than one module to be scheduled in non-overlapping periods.
- Different modules handled by different lecturers which doesn’t conflict can be allotted at the same period on same day.

Some Implicit Restraints:

- There are four timeslots for 2 hours lecture class: 8: 30 - **10:30, 10:35- 12:35,13:45-15:45and 16:00 - 18:00 for** five days (Monday - Friday).
- There are six slots for lecture class 8:45 - 9:45, 9:50 - 10:50, 10:55-11:55, 13:45-14:45 ,14:50-15:50 and 15:55 – 16:55 for five days (Monday - Friday)
- Lecture classes should be in any 2 consecutive days in a week.
- Tutorial classes with multiple cohorts should be in another 2 or 3 consecutive days in a week.
- Allow a one-hour lunch break to each intake students.
- Allow one – day free to each intake students between Monday - Friday.

Methodology 2

APU TTSS is developed to allocate classrooms and timeslots for both Lecture and Tutorial which satisfy the constraints imposed. Graph coloring algorithms can be customized to fit to any Lecture and Tutorial class problems for the minimum of modules specified. Various timeslots combinations can be acquired so that another Intake timetable scheduling is generated as of need.

Input data set: Combination of modules for lecturers with specialization.

Output data set: The minimum number of conflict free time slots required to schedule modules for both lecture (L) and tutorial (T) classes.

C. Bipartite Graph Algorithm

Step 1: Identify the list of lecturers and modules
Step 2: Generate adjacency matrix based on lecturers and modules
Step 3: Input the bipartite graph G.
Step 4: Calculate the degree sequence of each lecturer for the bipartite graph G.
Step 5: Assign color 1 to the vertex of G having highest degree.
Step 6: Assign color 2 to all non-adjacent uncolored vertices of v and store into Colored_Array.
Step 7: Assign new color, color 2 which is not previously used to the next non- colored vertex having highest degree.
Step 8: Assign color 2 to all non-adjacent uncolored vertices of v and store into Colored_Array.
Step 9: Repeat step 6 – 7 until all vertices are appropriately colored.
Step 10: Set total number of colors used in Colored_ array is equal to minimum number of non-conflicting timeslots.
Step 11: End

III. FINDINGS AND DISCUSSION

A. Optimal Solution 1

In this paper, graph coloring algorithm was used to design the solution. First draw an adjacency matrix having 10 rows and 10 columns corresponding to 9 modules shown in Table 2. The “1” in adjacency matrix indicates that corresponding row and column modules are adjacent to each other and “0” indicates that corresponding row and column modules are not adjacent to each other.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

	IMT	BCS	CITW	FEP	QSS	ISMT	QSM	FA	QA
IMT	-	1	1	1	1	1	1	1	1
BCS	1	-	1	1	1	1	1	1	1
CITW	1	1	-	1	1	1	0	0	0
FEP	1	1	1	-	1	0	1	1	0
QSS	1	1	1	1	-	1	0	1	1
ISMT	1	1	1	0	1	-	0	0	0
QSM	1	1	0	1	0	0	-	1	0
FA	1	1	0	1	1	0	1	-	1
QA	1	1	0	0	1	0	0	1	-

Table 2: Adjacency Matrix

The vertices of a graph to be colored using a set of *p* colors so that no two vertices are connected by an edge are assigned the same color. The minimum number of colors required to color the vertices (denoted by χ) is called the chromatic number of the graph. The problem of computing the chromatic number of a graph is NP-Complete. Here is a straightforward method that can be followed to color the vertices of a graph. Associate colors with the positive integers 1, 2, 3, …. For example, perhaps 1 is red, 2 is green, 3 is yellow, 4 is pink , 5 is purple and so forth. Choose the uncolored vertex that is adjacent to the greatest number of *colors*. If there is a tie, break the tie by choosing the vertex that is adjacent to the greatest number of *uncolored vertices*. If there is still a tie, break the tie arbitrarily. Color the chosen vertex with the first color (that is, the color associated with the smallest integer) that is not already used for one of the neighbors of the chosen vertex. Therefore, the number of colors required is shown in Table 3.

Module	IMT	BCS	CITW	FEP	QSS	ISMT	QSM	FA	QA
Color Number	1	2	3	4	5	4	3	6	3
Color	Red	Green	Yellow	Pink	Purple	Pink	Yellow	Blue	Yellow

Table 3: Number of colors required

Therefore, there are 6 unique colors that we have been able to find out to color the particular graph. In other words, the minimum number of timeslots required to conduct all 9 modules is 6. We can also draw colored conflict graph to find the number of timeslots required to conduct the classes for the specified 9 modules.

B. Course Conflict Graph

Considering each module as a vertex, edge between two vertices is drawn only if there is a common student(s). Given the conflicts between the modules (i.e., some modules can't be scheduled in the same timeslot) we allocate timeslots for the lectures and tutorials for the modules in different days. For level 1, all lecture (L) classes are two hour slots and tutorial (T) classes are one hour slots in Figure 1.

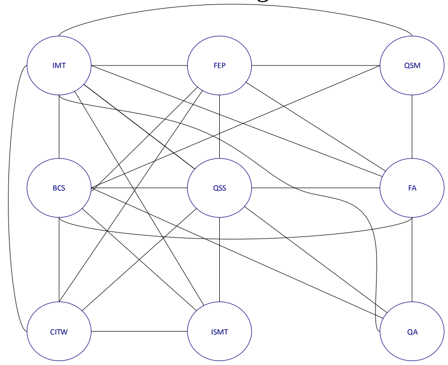

Figure. 1: Course Conflict graph (uncolored)

The minimum number of non-conflicting timeslots of the 9 modules is 6. After applying Graph coloring algorithm, the resultant graph in Figure 2and Figure 3 is properly colored for both lecture and tutorial classes with chromatic number 6.

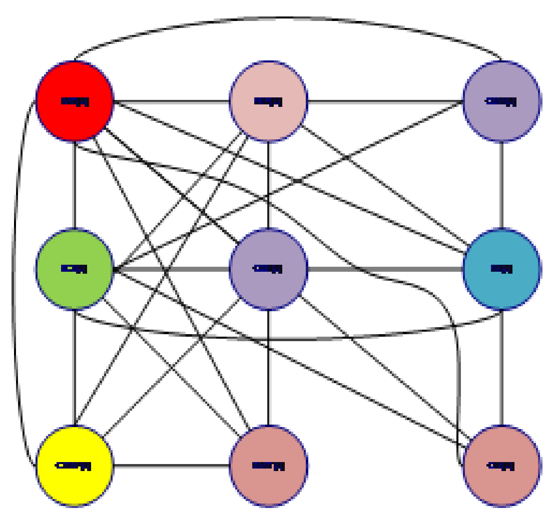

Figure 2: Colored Conflict Graph (L)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

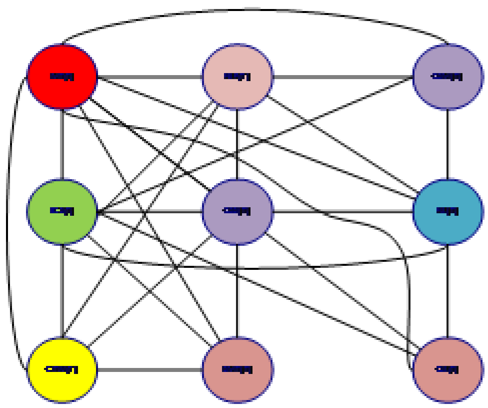

Figure 3: Colored Conflict Graph (T)

In the above solution, the minimum number of timeslots required is 6 which we can breakdown into two consecutive days for lecture classes and one day for tutorial classes.

C. Optimal Solution 2

Referring to the given problems, the number of “Lecture” and “Tutorial” classes for each module needed by a particular Lecturer is illustrated in Table 4.

	L1	L2	L3	L4
IMT	1	0	0	0
BCS	1	0	0	0
CITW	0	0	0	1
FEP	0	0	1	0
QSS	0	1	0	0
ISMT	0	0	1	0
QSM	0	1	0	0
FA	0	0	1	0
QA	1	0	0	0

Table 4: Module – Lecturer Requirement Matrix

The optimal solution is obtained through by proper edge coloring of the bipartite graph as shown in Figure 4. The study apply bipartite graph methodology to the data as in Table 4, the following graph is generated as in Figure 4.

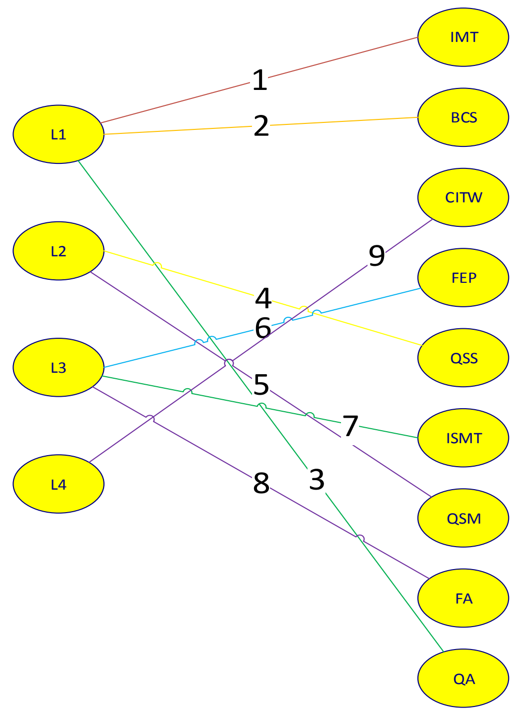

Figure 4: Bipartite Graph for both Lecturers& Modules

The finding in form of bipartite graph in Figure 4for both Lecture (L) and Tutorial (T) has been plotted in the table form. The optimal solution is illustrated in Table 5.1andTable 5.2.

Day	Time	Module	Lecturer
Monday	8:30-10:30	IMT	L1
	10:35-12:35	L4-QSS	L4
	12:35-1:45	LUNCH BREAK	
	13:45-15:45	QA/ISMT	L1/L3
Tuesday	8:30-10:30	QSM/FA/CITW	L2/L3/L4
	10:35-12:35	BCS	L1
	12:35-1:45	LUNCH BREAK	
	13:45-15:45	FEP	L3

Table 5.1: Final Lecture- Module timetable scheduling

Day	Time	Module	Lecturer
Wednesday	8:45-9:45	IMT	L1
	9:50-10:50	L4-QSS	L4
	10:55-11:55	QA/ISMT	L1/L3
	12:35-1:45	LUNCH BREAK	
	13:45-14:45	QSM/FA/CITW	L2/L3/L4
	14:50-15:50	BCS	L1
	15:55-16:55	FEP	L3

Table 5.2: Final Tutorial- Module timetable scheduling

The Table 5.1 and Table 5.2 show the list of each session in hourly intervals and displays the details of duration of lecture/ tutorial, lecturer name and related modules. On the completion of details it can generate timetable for a combination of maximum three different module allocations in a specified time on a day.

IV. CONCLUSION

The study has presented an appropriate method of timetable scheduling in APU TTSS complex problem by applying graph coloring technique. The study APU TTSS is developed in two phases. In the first phase, a module requirement matrix with graph coloring technique to construct the course conflict graph from the list of programs and combination of modules. In the second phase, a bipartite graph coloring is used to assign the modules to lecturers for both Lecture and Tutorial. The study develop a feasible solution from these two phase to overcome the complexity of APU TTSS. An optimal solution was derived from this study to overcome the problem of allocating lecturers to respective constraints such as modules, lectures and tutorial within the minimum timeslots without any conflicts. In future, this approach can be demonstrated for more complex problems to cater for additional variables such as class rooms, auditorium, laboratories, and so on.

V. EVIDENCE OF GRATIFICATION

The specified constraints are fulfilled by the above "Final Lecturer – Module timetable scheduling". There is no overlapping between modules and lecturers. There is no duplicate module / lecturer in any particular timeslot.

REFERENCES

1. Ayanegui, H. and Aragon.A.C (2009). A complete algorithm to solve the graph-coloring problem, 533(09), 107-108.
2. Assi, M., Halawi, B. and Haraty, R.A. (2018). Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem. Procedia Computer Science, 126, 899-906.
3. Broder.S, (1964) "Final Examination Scheduling," Communications of the ACM, 7(8).
4. Desroches, S., Laporte, G. and Rousseau, J.M, (1978) "HOREX: A computer program for the construction of examination". INFOR, 16(3), 294-298.
5. Joseph M.M. and Enokela, J.A (2012). Implementation of a Time Table Generator Using Visual Basic.Net, *ARPN Journal of Engineering and Applied Sciences*, 7(5), 1.
6. Matula, D.W., Marble. G, and Isaacson. D, (1972) "Graph Coloring Algorithms," in Graph Theory and Computing, R. Read (ed.), Academic Press, New York, 109-122.
7. Malkawi, M., Hassan, A and Osama, A.M (2008). *The International Arab Journal of Information Technology*: A New Exam Scheduling Algorithm Using Graph Coloring 5(1), 80-81.
8. Mittal, D., Doshi, H., Sunasra, M. and Renuka, N. (2015). *International Journal of Advanced Research in Computer and Communication Engineering:* Automatic Timetable Generation using Genetic Algorithm, 4(2), 1-3
9. Runa, G. and Siddhartha R. (2017), *International Journal of Computational and Applied Mathematics:* A Study on Course Timetable Scheduling using Graph Coloring Approach, 12(2), 469-485.
10. Saritha, M., Pranav, K.V., Pradeep, Mahesh, N.R (2017). Automatic Time Table Generator, *International Journal of Advanced Research in Computer Science and Software Engineering*, 7(5), 1.
11. Vinod, J.K, Samir, S. Y (2016). Academic Timetable Scheduling:Revisited, *International Journal of Research in Science & Engineering*, (16), 2.
12. Werra, D.D., (1985) "An introduction to timetabling", in *European Journal of Operational Research*, 19, 151-162.

AUTHORS PROFILE

Sireesha Prathi Gadapa is working as School of Mathematics, Accounting and Quantitative Studies, Asia Pacific University, Malaysia.

M. M. Mohamed Ubaidullah is working as School of Mathematics, Accounting and Quantitative Studies, Asia Pacific University, Malaysia

Rajasegeran Ramasamy is working as School of Mathematics, Accounting and Quantitative Studies, Asia Pacific University, Malaysia.

Raja Rajeswari Ponnusamy is working as School of Mathematics, Accounting and Quantitative Studies, Asia Pacific University, Malaysia.