International Journal of Computational and Applied Mathematics.
ISSN 1819-4966 Volume 12, Number 2 (2017), pp. 469-485

http://www.ripublication.com

A Study on Course Timetable Scheduling using Graph Coloring Approach

Runa Ganguli[1] and Siddhartha Roy[2]

*[1]Department of Computer Science,
The Bhawanipur Education Society College, Kolkata,West Bengal, India
Email runa.ganguli@gmail.com[1]*

*[2] Department of Computer Science,
The Bhawanipur Education Society College, Kolkata,West Bengal, India
Email siddhartha_r24@yahoo.com[2]*

Abstract

In any educational institution, the two most common academic scheduling problems are course timetabling and exam timetabling. A schedule is desirable which combines resources like teachers, subjects, students, classrooms in a way to avoid conflicts satisfying various essential and preferential constraints. The timetable scheduling problem is known to be NP Complete but the corresponding optimization problem is NP Hard. Hence a heuristic approach is preferred to find a nearest optimal solution within reasonable running time. Graph coloring is one such heuristic algorithm that can deal timetable scheduling satisfying changing requirements, evolving subject demands and their combinations. This study shows application of graph coloring on multiple data sets of any educational institute where different types of constraints are applied. It emphasizes on degree of constraint satisfaction, even distribution of courses, test for uniqueness of solution and optimal outcome. When multiple optimal solutions are available then the one satisfying maximum preferential conditions is chosen. This paper solely focuses on College Course Timetabling where both hard and soft constraints are considered. It aims at properly coloring the course conflict graph and transforming this coloring into conflict-free timeslots of courses. Course Conflict graph is constructed with courses as nodes and edges drawn between conflicting courses i.e. having common students.

Keywords: Graph Coloring, Course Timetable Scheduling, Hard Constraints, Soft Constraints, Course Conflict Graph

1.1 Basic Concepts of Graph Theory:

A graph G is an ordered triplet (V(G), E(G), ф) consisting of a non-empty set V of vertices or nodes, E is the set of edges and ф is the mapping from the set of edges E and the set of vertices V.

In a b**ipartite graph** (or **bigraph**) (Fig. 1) the set of vertices V can be partitioned into two disjoint sets *V1* and *V2* such that every edge of the graph connects a vertex in *V1* to one in *V2*, i.e. *V1* and *V2* are independent sets.

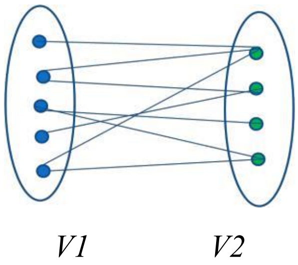
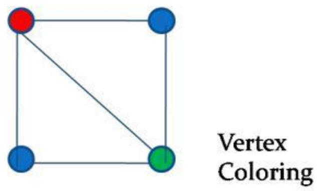


Fig. 1 Bipartite graph

Given a graph G, a vertex coloring of G (Fig. 2(a)) is a function f: V →C where V is a set of vertices of the graph G and C is the set of colors. It is often both conventional and convenient to use numbers 1, 2, ….n for the colors. Proper k-coloring [4][5] of G is a coloring function f which uses exactly k colors and satisfies the property that f(x) ≠ f(y) whenever x and y are adjacent in G.

The smallest number of colors needed to color G is known as it's chromatic number X(G)[4][5]. A graph that can be assigned a proper k-coloring is called k-colorable, and it is k-chromatic if it's chromatic number is exactly k. The chromatic polynomial counts the number of ways a graph can be colored using no more than a given number of colors.

Edge-coloring of graphs (Fig. 2(b)) is similar to vertex-coloring. Given a graph G, an edge-coloring of G is a function f_0 from the edges of G to a set C of elements called colors.



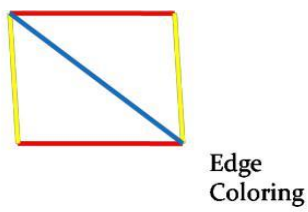


Fig. 2(a) Vertex Coloring **2(b) Edge Coloring**

We have studied that for every edge coloring problem there exists an equivalent vertex coloring problem [4] of its line graph. Given a graph G, its line graph L(G) is a graph such that each vertex of L(G) represents an edge of G, vertices of L(G) are adjacent if and only if their corresponding edges share a common endpoint (are incident) in G. The line graph L(G) of a given graph G is a simple graph whose proper vertex coloring gives a proper edge coloring of G by the same number of colors.

After discussing the introductory concept of graph theory related to graph coloring problem, in the above subsection 1.1, we present the literature survey in section 2, which describes various research works that have been done on scheduling problems using graph coloring method. The concept of scheduling problems in general along with various constraints involved and special emphasize on Course Timetable scheduling are mentioned in Section 3. Next Section 4 covers the working methodology and the corresponding algorithm used. Two cases of course timetable scheduling problems (Honours/Major-General/Minor Subject combination and Teacher-Subject Combination) of undergraduate courses under Indian Universities have been studied and corresponding solutions proposed. This paper ends in section 6 with some concluding remarks.

2. LITERATURE SURVEY

Solving timetabling problems through application of computers has a long and varied history. In 1967, the problem of course scheduling was applied to graph coloring [6][13][17][18]. In 1967 Welsh and Powell [10] illustrating the relationship between timetabling and graph coloring, and developed a new general graph coloring algorithm to solve (or approximately solve) the minimum coloring problem more efficiently. They were also successful in coloring graphs that arise from timetabling problems, more specifically examination timetabling problems. In 1969, Wood's graph algorithm [14] operated on two $n \times n$ matrices, where n denotes the number of vertices in the graph; a conflict matrix C was used to illustrate which pairs of vertices must be colored differently due to constraint restrictions in the problem and a similarity matrix S was used to determine which pairs of vertices should be colored the same. Dutton and Bingham in 1981 introduced two of the most popular heuristic graph coloring algorithms. Considering each color one by one, a clique [4] is formed by continually merging the two vertices with the most common adjacent vertices. On completion, identical coloring is applied to all the vertices which are merged into the same.

In 1986, Carter [19] in his examination timetabling survey refers to some of the above-mentioned graph coloring algorithms and heuristics and shows how graph theoretic approach to timetable scheduling is one of the most popular. It has been accepted and applied by many educational institutions to solve their examination timetabling problems. According to Carter in his survey, work of Mehta is significant as its objective of obtaining "conflict-free" schedules, given a fixed number of time periods turned out to be one of the most complex timetabling applications [15]. In

1991, Johnson, Aragon, McGeoch and Schevon [20] implemented and tested three different approaches for graph coloring with a simulated annealing technique, observing that simulated annealing algorithms can achieve good results, but only if allowed a sufficiently large run time. In 1992, Kiaer and Yellen in a paper [21] describes a heuristic algorithm using graph coloring approach to find approximate solutions for a university course timetabling problem. The algorithm using a weighted graph to model the problem aimed at finding a least cost k-coloring of the graph (k being number of available timeslots) while minimizing conflicts. In 1994, Burke, Elliman and Weare [22] introduced plans for a university timetabling system based on graph coloring and constraint manipulation. Graph coloring and room allocation heuristic algorithms were described along with an illustration of how the two can be combined to provide the basis of a system for timetabling. The authors also discussed the handling of several common timetabling features within the system, primarily with regards to examination timetabling. In 1995, graph coloring method was introduced aiming at optimizing solutions to the timetable scheduling problems [11]. Bresina (1996) was among the early researchers who used this approach and made several modifications in the manual approach conducted at universities [23]. In 2007, for university timetabling an alternative graph coloring method was presented that incorporates room assignment during the coloring process [6]. In 2008, the Koala graph coloring library was developed which includes many practical applications of graph coloring, and is based on C++ [7]. In 2009, automata-based approximation algorithms were proposed for solving the minimum vertex coloring problem [8]. A. Akbulut and G. Yılmaz in 2013[24] proposed a new university examination scheduling system using graph coloring algorithm based on RFID technology. This was examined by using different artificial intelligence approaches. Also, in recent year researchers have been exploring new alternative methods to deal scheduling problems for obtaining better result.

3. SCHEDULING PROBLEM

The general idea of a scheduling problem is defined by allocation of related resources among a number of time-slots satisfying various types of essential and preferential constraints aiming at creating optimized conflict-free schedule.

Some of the typical scheduling problems include-

- Timetable Scheduling

a. Course Timetable Scheduling- courses sharing common resources to be scheduled in conflict-free time-slots.

b. Exam Timetable Scheduling – exams sharing common resources to be scheduled in conflict-free time-slots.

- Aircraft Scheduling- aircrafts need to be assigned to flights.

- Job Shop Scheduling- for a given set of jobs with their processing times and a given set of machines, a schedule mapping jobs to machines meeting feasibility constraints and optimization objectives.

3.1 Motivation towards Timetable Scheduling

Effective timetable is vital to the performance of any educational institute. It impacts their ability to meet changing and evolving subject demands and their combinations in a cost-effective manner satisfying various constraints. In this paper, we have focused our work into Course Timetable Scheduling.

3.2 Course Timetable Scheduling

Course Timetabling is the scheduling of a set of related courses in a minimal number of time-slots such that no resource is required simultaneously by more than one event. In a typical educational institute resources, which may be required by courses simultaneously can be students, classrooms and teachers.

3.3 Constraints

Constraints are the most vital aspect of any scheduling problem. These are the various restrictions involved in creating a schedule. Based on satisfaction of these a schedule can be accepted or get rejected. Depending on the degree of strictness, constraints are broadly classified into- Hard and Soft Constraints [11].

3.3.1 Hard and Soft Constraints

Hard Constraints are those essential conditions which must be satisfied to have a legal schedule. If any of the hard constraints cannot be placed successfully by a schedule, then such a schedule is rejected. For example, no two subjects having common students can be scheduled in the same time-slot, courses cannot be assigned to more than maximum number of available time-slots or periods. In those scheduling datasets which involve resources like teachers and classrooms, no courses can be scheduled to the same classroom at same time-slot, more than one course taught by the same teacher cannot be assigned same time of the week.

Soft Constraints on the other hand are those preferential conditions which are optional. Mostly, it gets difficult to incorporate all the soft constraints in a schedule. A schedule is still said to be legal even if it fails to satisfy soft constraints, provided all hard constraints are met. For example- a teacher may prefer to take practical classes only in the second half, honours and pass classes are preferred to be scheduled in non-overlapping time-slots, etc.

3.3.2 Temporal and Spatial Constraints

Constraints can also be viewed as time-related and space-related conditions. Time-related constraints are called Temporal constraints. For example, Computer Science theory and practical classes cannot be scheduled at the same time-slot or period because of common students. Also, there must be a fixed number of theory and practical classes scheduled in a week.

On the other hand, space-related constraints are called Spatial constraints. In course scheduling problem, spatial constraints mainly involve classroom related issues. Any educational institute has a fixed number of available rooms with specified capacity. Also, classrooms can be theory or laboratory based. While making schedules, courses having student capacity compatible to the classroom size is an essential condition. Courses which need specific classroom type need to be assigned accordingly.

Both temporal and spatial constraints are mainly hard type constraints whose fulfilment determines the effectiveness of a schedule.

4. METHODOLOGY

For solving Course Timetable scheduling problems using graph coloring, the problem is first formulated in the form of a graph where courses act as vertices. Depending on type of graph created, edges are drawn accordingly. One type is conflicting graph where edges are drawn between conflicting courses having common students. Other one is non-conflicting graph, where edges are drawn between mutually exclusive courses having no students in common. Sometimes it is found that creating a non-conflicting graph from the given input set and constraints is easier costing less time. This non-conflicting graph needs to be complemented to obtain the required conflict graph whose proper coloring provides the desired solution. This two-step method is efficient in certain cases, while in some conflict graph is created directly.

Some problems involve few resources while others may require many at a time. Courses can also conflict due to common teachers, common classrooms in addition to common students. In such cases, the conflict graph must consider course, teacher and room conflicts simultaneously.

As mentioned earlier, there can be various aspects of a scheduling problem. When teachers are involved in resources, other factors like availability of teachers, subject area preferred by each teacher acts as additional data inputs which needs to be provided for making a complete schedule.

4.1. Graph Coloring Algorithm:

Input: The course conflict graph G thus obtained act as the input of graph coloring algorithm.

Output: The minimum number n of non-conflicting time-slots required to schedule courses.

Degree sequence is the array having degree of each vertices of the input graph G. Colors being used are stored in Used_Color array. And the chromatic number will be the total number of elements in the Used_Color array.

Step 1: Input the conflict graph G.

Step 2: Compute degree sequence of the input conflict graph G.

Step 3: Assign color1 to the vertex vi of G having highest degree.

Step 4: Assign color1 to all the non-adjacent uncoloured vertices of vi and store color1 into Used_Color array.

Step 5: Assign new color which is not previously used to the next uncoloured vertex having next highest degree.

Step 6: Assign the same new color to all non-adjacent uncoloured vertices of the newly colored vertex.

Step 7: Repeat step-5 and step-6 until all vertices are colored.

Step 8: Set minimum number of non-conflicting time-slots n= chromatic number of the colored graph=total number of elements in Used_Color array.

Step 9: End

5. CASE STUDIES

Undergraduate colleges under Indian Universities offer a variety of subject combinations to its students. In streams like B.A or B.Sc., students can take one subject as Honours (Major) and two subjects as General (Minor/Pass) papers. Also, to conduct such courses teachers of respective subjects are needed to be scheduled according to their availabilities in minimum number of time slots without any conflict. In the following subsection, we have presented two such typical cases of scheduling problems and their conflict free solution timetables.

5.1 Example 1: Honours and General subject combination

Problem Definition:

Suppose in any College of undergraduate science students, given n number of honours subjects and m number of general subjects, the available number of p periods course timetable should be prepared satisfying some given constraints. The objective is to find minimum number of time slots to schedule all the courses without conflict.

Input Dataset: Table 1 shows the Honours-General subject combination of a typical undergraduate science course.

Table 1 Honours-General Subject Combination

Sl. No.	List of Honours Subjects	General Subject Combination
1	Physics	Mathematics(compulsory) + Computer Science/Chemistry/Electronics
2	Chemistry	Mathematics(compulsory) + Physics/Computer Science
3	Mathematics	Physics(compulsory) + Chemistry/Computer Science/Statistics
4	Economics	Mathematics(compulsory) + Statistics/Computer Science

List of constraints:

Hard Constraints-

- Courses having common student cannot be allotted at the same time slot on the same day.
- Total number of available periods is 8. (maximum)

Soft Constraints-

- Honours and General courses need to be scheduled in non-overlapping time-slots.

Solution:

Considering each course as a node, edge between two nodes is drawn only if there is common student. (See Fig. 3)

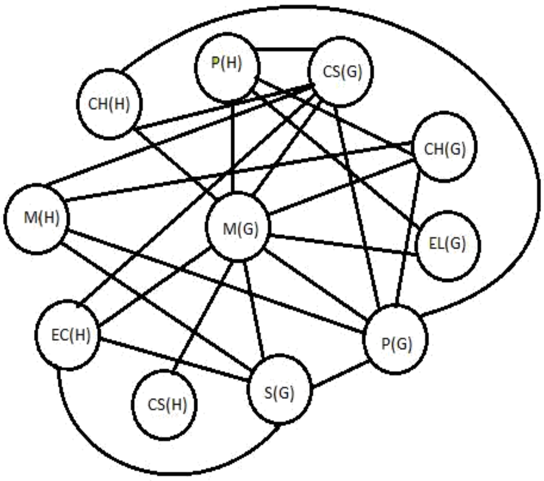


Fig. 3 Course Conflict Graph

N.B. (H) and (G) notations represent Honours and General courses respectively.

CH- Chemistry, M- Mathematics, EC-Economics, CS-Computer Science, P-Physics, S-Statistics, EL-Electronics.

After applying graph coloring algorithm, the resultant graph in Fig. 4 is properly colored with chromatic number 4. This is the minimum number of non-conflicting time-slots scheduling all the given courses.

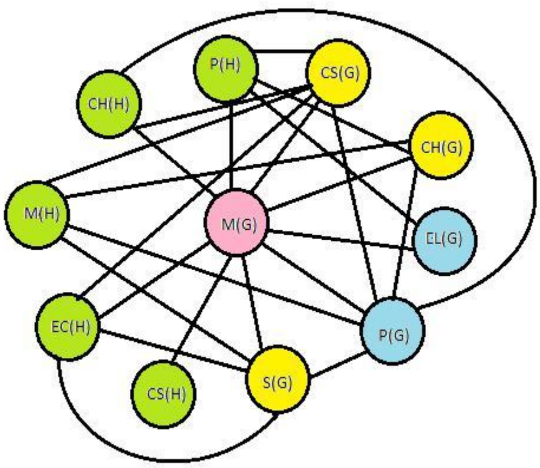


Fig. 4 Colored Conflict Graph

Result:

In the above solution, the hard constraints are satisfied properly. The resultant minimum number of time slots needed is 4 which do not exceed the total available 8 periods. All the Honours subjects cannot have common student, hence allotted in slot-1. Similarly, no student overlapping is possible between Electronics and Physics General, or students between Chemistry, Computer Science and Statistics General.

Mathematics(General) being compulsory for all Honours subject except Mathematics itself, is given separate slot 2. But since there can never be common students in Mathematics(H) and Mathematics(G), alternative solutions exist. Similarly, there can be other legal combinations as well.

Alternative Solutions

Using the minimum number of colors i.e. 4 for the above problem, there are many alternative legal solutions. Some of them are shown in Fig 5. All of these scheduled courses are non-conflicting combinations.

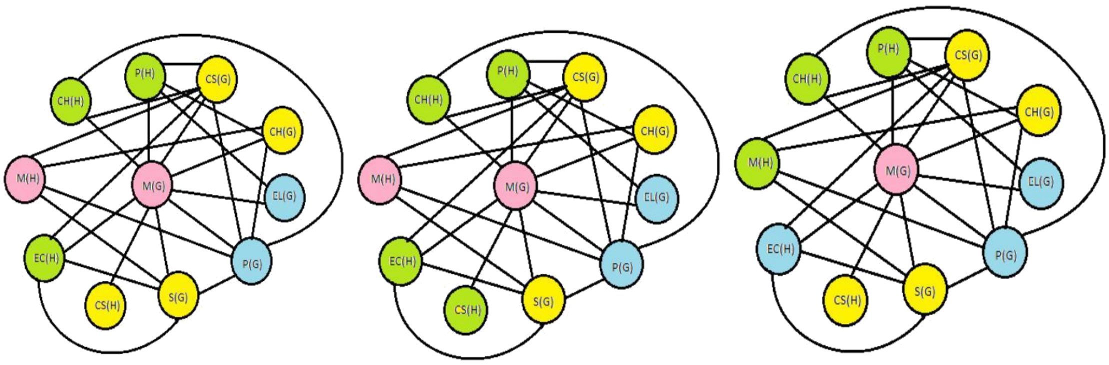


Fig. 5 Alternative Solutions

5.2 Example 2: Teacher-Subject problem

Problem Definition:

For a given ‘T’ number of teachers, ‘N’ number of subjects and available ‘P’ number of periods, a timetable should be prepared. The number of classes for each subject needed by a particular teacher is given in Table 2. This problem is mentioned earlier in some papers [25] [26] but only a partial solution was provided in both.

Input Dataset:

Number of teachers- 4

Number of subjects- 5

Table 2 Teacher-Subject Requirement Matrix

Periods P	N1	N2	N3	N4	N5
T1	2	0	1	1	0
T2	0	1	0	1	0
T3	0	1	1	1	0
T4	0	0	0	1	1

List of constraints:

Hard Constraints-

- At any one period, each subject can be taught by maximum one teacher.
- At any one period, each teacher can teach at most one subject.

Soft Constraints-

- Teacher taking two classes of same subject to be scheduled in consecutive periods.
- No more than two consecutive theory classes can be assigned to same teacher for teaching same subject.

Solution:

This problem is solved using bipartite graph which acts as the conflict graph. The set of teachers and subjects are the two disjoint independent sets. Edges are drawn connecting a vertex from Teacher set to a vertex in Subject set, indicating that the subject is taught by the respective teacher. This data is obtained from the Teacher-Subject requirement matrix given in Table. 2. The solution to the problem is obtained by proper edge coloring of the bipartite graph as shown in Fig. 6. The chromatic number acts as the minimum number of periods.

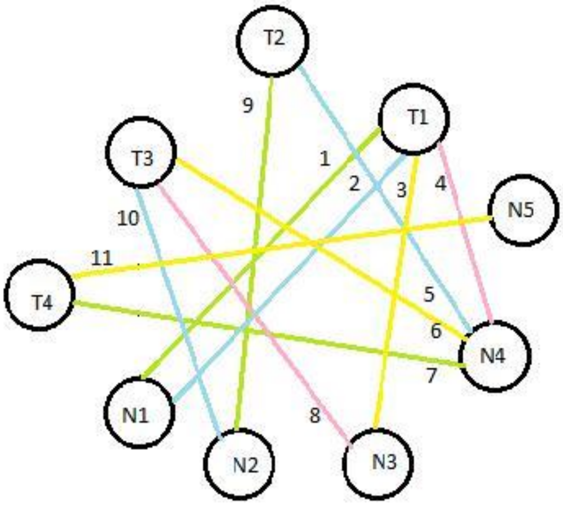


Fig. 6 Bipartite Graph G

An alternative way of solving the above problem is by converting the edge coloring problem into a vertex coloring problem. For that the bipartite graph is converted into its equivalent line graph L(G) [4], and a proper vertex coloring of the line graph shown in Fig. 7 gives the same solution. This is simply an alternative procedure and is used depending on the type of graph coloring needed to be applied. The eleven edges present in the bipartite graph in Fig. 6 acts as the vertices of L(G) in Fig. 7.

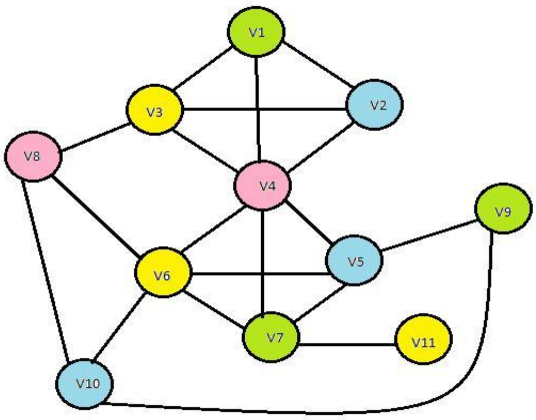


Fig. 7 Line graph L(G)

Result:

Graph Coloring of the line graph L(G) in Fig. 7 has been plotted in the solution Table 3 below.

Table 3 Graph Coloring Solution Table of Fig. 7

PINK	GREEN	BLUE	YELLOW
V4	V1	V2	V3
V8	V7	V5	V6
	V9	V10	V11

Now, each of these colors in Table 3 represents periods in Table 4 and the vertex in L(G) (Fig. 7) that corresponds to a particular edge in the bipartite graph G (Fig. 6) represents the teacher-subject combination scheduled under that period.

Thus, the final complete schedule is obtained and shown in Table 4.

Table 4 Final Teacher-Subject Allotment Table

Period 1	Period 2	Period 3	Period 4
T1-N4	T1-N1	T1-N1	T1-N3
T3-N3	T4-N4	T2-N4	T3-N4
	T2-N2	T3-N2	T4-N5

Proof of satisfaction:

It can be easily established that the specified constraints are satisfied by the above schedule.

- No common subject in any column indicates that at any particular period, a subject is taught by only one teacher.
- No duplicate data in any cell indicates that at any particular period, a teacher can teach only one subject.

It is known that the number of classes that can run simultaneously depends on the number of available teachers. Here, in the above example there are maximum 3 allocations in any column i.e. maximum three classes can run parallelly.

6. CONCLUSION

The complexity of a scheduling problem is directly proportional to the number of constraints involved. There is no fixed algorithm to solve this class of problem. Here we have studied a typical honours(major) and general(minor) course combination scheduling problem under university curriculum. Uniqueness and optimality are the main concerns in this scheduling. For the same chromatic number, there are many alternative solutions, and thus it is not unique. Although all the solutions can be claimed optimal when solved using minimum number of colors, a better schedule is one which maximizes satisfaction of soft constraints among its alternative solutions. In addition, we have also studied a teacher-subject scheduling problem where two alternative graph coloring methods (edge coloring using bipartite graph and vertex coloring using line graph) were applied and a complete solution is provided. The dynamic nature of scheduling problem challenges to further experiment with large data sets and complex constraints. An algorithm which can evenly distribute resources among available time-slots without conflict, create unique and optimized schedule and satisfy all hard and maximum number of soft constraints can be called ideal. Finding such algorithm is surely an evolving area of further research.

REFERENCES

[1] Kenneth, A., and Wolfgang, H., 1977, "Every Planar Map is Four Colorable. I. Discharging," Illinois Journal of Mathematics, 21(3), pp. 429–490.

[2] Heawood, P. J., 1980, "Map-Colour Theorems, 1980," Quarterly Journal of Mathematics, Oxford, 24, pp. 332–338.

[3] Birkhoff, G.D., 1912, "A determinant formula for the number of ways of coloring a map," Annals of Mathematics, 14, pp. 42-46.

[4] Deo, N., 1990, "Graph theory with applications to engineering and computer science," Prentice Hall of India.

[5] Bondy, J. A., 1969, "Bounds for the chromatic number of a graph," Journal of Combinatorial Theory, 7, pp. 96-98.

[6] Redl, T.A., 2007, "University Timetabling via Graph Coloring: An Alternative Approach," Congressus Numerantium, 187, pp. 174-186.

[7] Dobrolowski, T., Dereniowski, D., and Kuszner, L., 2008, "Koala Graph Coloring Library: An Open Graph Coloring Library for Real World Applications," Proceedings of the 2008 1st International Conference on Information Technology, IT, Gdansk, Poland, pp. 1-4.

[8] Torkestani, J.A., and Meybodi, M.R., 2009, "Graph Coloring Problem Based on Learning Automata," International Conference on Information Management and Engineering, pp. 718-722.

[9] Klotz, W., 2002, "Graph Coloring Algorithms," Mathematic- Bericht 5, TU Clausthal, pp. 1-9.

[10] Welsh, D.J.A., and Powell, M.B., 1967, "An Upper Bound for the Chromatic Number of a Graph and it's Application to Timetabling Problems," The Computer Journal. 10(1), pp. 85-86.

[11] Miner, S.K., Elmohamed, S., and Yau, H.W., 1995, "Optimizing Timetabling Solutions using Graph Coloring," NPAC, Syracuse University, pp. 99-106.

[12] Garey, R. D., and Johnson, S., 1979, "Computers and intractability: A Guide to the Theory of NP-Completeness," Freeman.

[13] Burke, E. K., Elliman, D. G., and Weare, R., 1993, "A university timetabling system based on graph coloring and constraint manipulation," Journal of Research on Computing in Education, 27(1), pp. 1-18.

[14] Wood, D. C., 1968, "A System for Computing University Examination Timetables," The Computer Journal, 11, pp.41-47.

[15] Karp, R. M., 1972, "Reducibility among Combinatorial Problems," in Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, Eds., The IBM Research Symposia Series, New York, NY: Plenum Press, 1972, pp. 85-103.

[16] Bondy, S., 2002, "Final Examination Scheduling," Communications of the ACM, 22(7), pp. 494-498.

[17] Peck, J. E. L., and Williams, M. R., 1966, "Examination Scheduling," Communications of the ACM, 9(6), pp. 433-434.

[18] Wood, D. C., 1969, "A Technique for Coloring a Graph Applicable to Large Scale Timetabling Problems," The Computer Journal, 12, pp. 317-319.

[19] Carter, M. W., Laporte, G., and Lee, S. Y., 1996, "Examination timetabling: Algorithmic strategies and applications," Journal of the Operational Research Society 47(3), pp. 373–383.

[20] Johnson, D. S., Aragon, C. R., McGeoch, L. A., and Schevon, C., 1991, "Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning," Operations Research, 39(3), pp. 378–406.

[21] Kiaer, L., and Yellen, J., 1992, "Weighted Graphs and University Timetabling," Computers and Operations Research, 19(1), pp. 59-67.

[22] Burke, E.K., Elliman, D.G., and Weare, R.F., 1994, "A University Timetabling System Based on Graph Colouring and Constraint Manipulation," Journal of Research on Computing in Education, 27(1), pp. 1-18.

[23] Bresina, J., 1996, "Heuristic-Biased Stochastic Sampling," Proceedings of the 13th National Conference on Artificial Intelligence, pp. 271-278.

[24] Akbulut, A., and Yılmaz, G., 2013, "University Exam Scheduling System Using Graph Coloring Algorithm and RFID Technology," International Journal of Innovation, Management and Technology, 4(1), pp. 66-72.

[25] Sarma S. V. M., 2012, "Application of graph theory in human life," International Journal of Computer Application, 1(2), pp. 21-30.

[26] Shirinivas, S. G., Vetrivel, S., and Elango, N. M., 2010, "Application of Graph Theory in Computer Science An Overview," International Journal of Engineering Science and Technology, 2(9), pp. 4610-4621.