Available Online at http://www.recentscientific.com

CODEN: IJRSFP (USA)

International Journal of Recent Scientific Research
Vol. 9, Issue, 2(C), pp. 23939-23945, February, 2018

ISSN: 0976-3031

International Journal of Recent Scientific Research

DOI: 10.24327/IJRSR

Research Article

AN INSTRUCTION ON COURSE TIMETABLE SCHEDULING APPLYING GRAPH COLORING APPROACH

Narayan Poddar*[1] and Basudeb Mondal[2]

[1]Department of Mathematics, Bakshirhat High School, Bakshirhat, Coochbehar, West Bengal-736131, India
[2]Bajitpur Primary School, Bajitpur, Purapara, Murshidabad, West Bengal, 742201, India

DOI: http://dx.doi.org/10.24327/ijrsr.2018.0902.1567

ARTICLE INFO

Article History:

Received 17th November, 2017
Received in revised form 12th December, 2017
Accepted 04th January, 2018
Published online 28th February, 2018

Key Words:

Graph Coloring, Course Timetable Scheduling, Hard Constraints, Soft Constraints, Course Conflict Graph.

ABSTRACT

In any educational institution, the two most common academic scheduling problems are course timetabling and examination scheduling. A schedule is desirable which combines resources like teachers, subjects, students, classrooms, and teaching- learning aids in a way to avoid conflicts satisfying various essential and preferential constraints. The timetable scheduling problem is informed to be NP Complete but the similar optimization problem is NP Hard. Hence a heuristic approach is proposed to find a nearest optimal solution within reasonable running time. Graph coloring is one such heuristic algorithm that may 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 various types of constraints are applied. It emphasizes on degree of constraint satisfaction, even distribution of courses, test for uniqueness of solution and optimal conclusion. When multiple optimal solutions are available then the one satisfying maximum discriminating conditions is chosen. This paper only focuses on College Course Timetabling where both hard and soft constraints are considered. Its objectives at properly coloring the course conflict graph and transforming this coloring into conflict-free times, lots of courses. Course Conflict graph is constructed with courses as vertices and edges drawn between conflicting courses i.e. having common students.

INTRODUCTION

In the year 1736, graph theory originated from the Königsberg bridge problem pointed out by mathematician Euler which later known as Eulerian graph [9]. In the same decade, Gustav Kirchhoff established the idea of a tree, a connected graph without cycles which was used in the calculation of currents in electrical networks or circuits and later to enumerate chemical molecules. In 1840, A.F Mobius introduce the idea of complete graph and bipartite graph (section 1.1). In 1852, Thomas Guthrie introduce the famous four-color problem. The first results about graph coloring notonly with planar graphs in the form of the coloring of maps [2]. Even by the four-color problem was invented it was solved only after a century by Kenneth Appel and Wolfgang Haken [15]. In 1890, Heawood substantiated the five-color theorem, saying that every planar map may be colored with no more than five colors [12]. In 1912, George David Birkhoff to study coloring problems in algebraic graph theory initiated the chromatic polynomial [3][9]. Graph Coloring has lot of real-time applications including map coloring, time-table scheduling problem, parallel computation, network design, Sudoku, register allocation, bipartite graph detection, etc. [2][9]. Graph coloring has considerable application to a large variety of complex problems involving optimization [18] In particular, conflict resolution, or the optimal partitioning of mutually exclusive events, may often be accomplished by means of graph coloring. Examples of similar problems include course or examination timetable scheduling [6][21][26].

While constructing schedule of classes at a college or university, it is explicit that courses taught by the same professor and courses that require similar classroom must be scheduled at different time -period. Also, for a curriculum, a particular student or group of students may require to accept two different but related courses (e.g. Mathematicsand Physics) during a semester. In such cases, courses are necessarily being scheduled in a way to avoid conflicts. Thus, the problem of

Corresponding author:* **Narayan Poddar
Department of Mathematics, Bakshirhat High School, Bakshirhat, Coochbehar, West Bengal-736131, India

determining minimum or reasonable number of time periods that may successfully schedule all the courses subjected to the restrictions is a typical graph coloring problem [4][20][26]. This paper is connected with the problem of course timetable scheduling, where graph coloring may provide an algorithm [17] which will prevent or at least minimize conflicting schedules. So, optimal solutions to such problems may be found by determining minimal coloring for the corresponding graphs. Unfortunately, this may not always be accomplished in polynomial time. As the graph coloring problem is known to be NP-complete [9][11] there is no known algorithm which, for every graph, will optimally colored the nodes of the graph in a time bounded by a polynomial in the number of nodes. More than graph coloring algorithm such as the Saturation algorithm, the Recursive Largest First algorithm, Simulated Annealing algorithm, Greedy algorithm, are NP complete [9][11].

Basic Concepts of Graph Theory

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

A simple graph G= (V, E) is Bipartite if we may partition its vertex set V to disjoining sets U and V such that there are no edges between U and V. We say that (U, V) is a Bipartition of G. A Bipartite Graph is displayed in Figure-1.

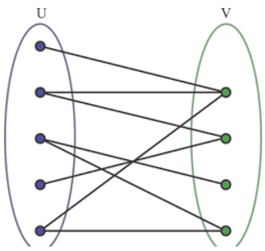

Figure 1 Bipartite graph

Given a graph G, a vertex coloring of G (Figure-2.1.) is a function ψ: 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 [9][3] of G is a coloring function ψ which uses exactly k colors and satisfies the property that $\psi(x) \neq \psi(y)$ whenever x and y are adjacent in G.

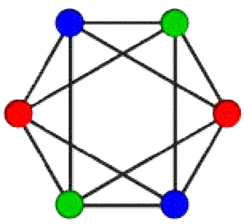

Figure 2 Vertex Coloring

The minimum number of colors require to color G is known as it's chromatic number X(G)[9][3]. A graph that assigns properly k-colors is known as k-colorable, and it is k-chromatic if its chromatic number is exactly k. The chromatic polynomial counts the number of ways a graph may be colored using no more than a given number of colors.

Edge-coloring of graphs (Figure-2.2.) is corresponding to vertex-coloring. Given a graph G, an edge-coloring of G is a function ψ_0 from the edges of G to a set C of elements called colors.

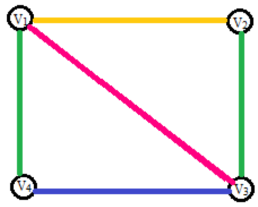

Figure 3 Edge Coloring

We know that for every edge coloring problem there exists an equivalent vertex coloring problem [9] of its line graph. Given a graph G, its line graph L(G) is a graph such that for every vertex of L(G) represents an edge of G, vertices of L(G) are adjacent if and only if their corresponding edges connect a common endpoint 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 equal number of colors.

After studying the basic concept of graph theory corresponding to graph coloring problem, in the above sub-article.1.1, we present the survey of literaturein article-2, which associates different research works that have been done on timetable scheduling problems using graph coloring method. The idea of scheduling problems in general along with different constraints associated and specially emphasized on Course Timetable scheduling are introducing in Article-3. In next Article-4, we discussed the working methodology and the corresponding algorithm used. Two cases of course timetable scheduling problems (Honours/Major-General/Minor Subject Combination and Subject-Teacher Combination) of undergraduate courses under Indian Colleges and Universities have been studied and corresponding solutions proposed. This paper ends in Article-6 with some concluding remarks.

Survey of Literature

Solving timetabling problems with the help of computer applications has a long and varied history. In 1967, the problem of course scheduling was emphasised to graph coloring [6][20][21][27]. In 1967, Welsh and Powell [25] illustrated the relationship between timetabling and graph coloring, and introduced 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 occur from timetabling problems, more specifically examination timetabling problems. In 1969, Wood's graph algorithm [26] acted on two n × n matrices, where n denotes the number of vertices in the graph; a conflict matrix C was used to illustrate which pairs of vertices should be colored differently for constraint restrictions to the problem and a similar matrix S was used to determine which pairs of vertices should be colored the same. In 1981, Dutton and Bingham introduced two of the most popular heuristic graph coloring algorithms. Considering each color one by one, a clique graph [9] is formed by continually merging the two vertices with the most common adjacent vertices. Finally, identical coloring is used to all the vertices which are merged into the same.

In 1986, Carter [8] in his examination timetabling survey refers to some of the said graph coloring algorithms and heuristics and proves how graph theoretical approach to timetable scheduling which is one of the most popular. It has been accepted and applied by many Educational Institutions to schedule their examination timetables. As Carter, in his survey, Mehta's works are applicable to obtain "conflict-free" schedules, to a fixed number of time periods is one of the most complex timetabling applications [14]. In1991, Johnson, Aragon, McGeoch and Schevon [13] applied three different ways of graph coloring with a simulated annealing technique, observing that simulated annealing algorithms may give the best results, but it takes a sufficiently large run time. In 1992, Kiaer and Yellen in their paper [16] illustrate a heuristic algorithm using graph coloring approach to find approximate results for a university course timetabling problem. The algorithm applying 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 We are [7] associated plans for a university timetabling system using graph coloring and constraint manipulation. Graph coloring and room allocation heuristic algorithms were used along with an illustration of how the two can be combined to provide the basis of a system for timetabling. The authors also illustrate to handling several common timetabling features within the system, such as examination timetabling. In 1995, graph coloring method used to get optimized solutions to the timetable scheduling problems [18]. In 1996, Bresina was in the early researchers who used this approach and made several modifications in the manual approach used at universities [5]. In 2007, Redl for university timetabling an alternative graph coloring method was presented that incorporates room assignment during the coloring process [21]. In 2008, the Koala graph coloring library was developed which includes many practical applications of graph coloring, and is based on C++ [10]. In 2009, J.A. Torkestani and M.R. Meybodi introduced automata-based approximation algorithms for solving the minimum vertex coloring problem [24]. In 2013 A. Akbulut and G. Yılmaz introduced a new university examination scheduling system using graph coloring algorithm based on RFID technology [1]. This was examined by using various artificial intelligence approaches. Also, now researchers have been exploring new alternative methods to solve scheduling problems for obtaining better result.

Scheduling Problem

The common thinking about scheduling problem is defined by allocation of related resources among a number of time-slots satisfying various types of important and preferential constraints acted at creating optimized conflict-free schedule. Some of the typical scheduling problems include-

Timetable Scheduling

Course Timetable Scheduling- courses with common resources to be scheduled in conflict-free time-periods.

Exam Timetable Scheduling - exams with common resources to be scheduled in conflict-free time-periods.

Aircraft Scheduling- aircrafts need to be assigned to flights.

Job Shop Scheduling- for a number of jobs with their acting times and a given number of machines, a schedule mapping from jobs to machines assign feasibility constraints and optimization objectives.

Motivation towards Timetable Scheduling

Effective timetable is the most important for any educational institute to perform the best. Applying this they can change and evolve subject demands and their combinations in a cost-effective manner satisfying various constraints. In this paper, we have worked into Course Timetable Scheduling.

Course Timetable Scheduling

Course Timetabling is the scheduling of a set of related courses in a minimal number of time-periods such that no resource is required simultaneously by more than one event. There arise students, classrooms and teachers simultaneously as important resources in a typical educational institution.

Constraints

Constraints play the most important role in any scheduling problem. These are the different restrictions used to create a schedule. Based on satisfaction of these a schedule may be accepted or rejected. Depending on the degree of strictness, constraints are preferably classified into- Hard and Soft Constraints [18].

Hard and Soft Constraints

Hard Constraints are those essential conditions which must be satisfied to have a legal schedule. If one 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-period, courses cannot be assigned to more than maximum number of available time-periods. In those scheduling datasets which associate resources as teachers and classrooms, no courses can be scheduled to the same classroom at same time-period, more than one course taught by the same teacher cannot be assigned same time of the week.

On the other hand, *Soft Constraints* 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 satisfied. As an 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-periods, etc.

Temporal and Spatial Constraints

Some Constraints can also be used as time-related and space-related conditions. Time-related constraints are called *temporal constraints*. Such as, in Computer Science classes, theory and practical classes cannot be scheduled at the same time-period as there are common students. Also, there must be associate a fixed number of theory and practical classes in a week.

A *space-related constraints* are called *spatial constraints*. In course scheduling problem, spatial constraints mainly involve classroom related mater. 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 accommodation is an essential condition. Courses which need specific classroom have to be assigned accordingly.

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

METHODOLOGY

To solve such Course Timetable scheduling problems with the help of graph coloring, the problem is to formulate first in the form of a graph where courses act as vertices of the graph. The edges are drawn depending on the type of the graph accordingly. One of which is conflicting graph in where edges are drawn between conflicting courses having common students. Other is non-conflicting graph, whose edges are drawn between mutually exclusive courses having no students in common. Often it is found that creating a non-conflicting graph from the given inputs and constraints is easier and costing less time. This type of non-conflicting graph requires to be complemented to get the desired conflict graph whose proper coloring provides the required solution. This two-steps method is efficient in few cases, while in some conflict graph is created directly.

Some problems associate with few resources while others may require many at a time. Courses can also conflict due to common teachers, common classrooms into 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.

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-periods required to schedule courses. Degree sequence is the array having degree of each vertices of the input graph G. Used colors are stored in Used_Color array. And the chromatic number 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 v_i of G having highest degree.
Step 4: Assign color1 to all the non-adjacent uncoloured vertices of v_i 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-period n= chromatic number of the colored graph=total number of elements in Used-Color array.
Step 9: End

Case Studies

Colleges and Universities offer a variety of subject combinations to their undergraduate students. Students can take one subject as Honours (Major) and two subjects as General (Minor/Pass) in bachelor degree like B.A. or B.Sc. So to go through such courses they need teachers of respective subjects to schedule according to their availabilities in desirable number of time periods without any conflict. We have presented two such cases of scheduling problems with their conflict free solution timetables in the next sub- article.

Example: - Honours and General subject combination

Problem Definition: Suppose undergraduate science students have given m number of honours subjects and n number of general subjects in a college. The available number of p periods course timetable should be prepared for them satisfying some given constraints. The objective is to find least number of time periods to schedule all the courses without conflict.

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

Table 1 Honours-General Subject Combination

Serial No.	List of Honours/Major Subjects	General/Minor Subjects Combinations
1.	Mathematics	Physics(compulsory) + Chemistry/Computer Science/Statistics
2.	Physics	Mathematics(compulsory) + Computer Science/Chemistry/Electronics
3.	Chemistry	Mathematics(compulsory) + Physics/Computer Science
4.	Economics	Mathematics(compulsory) + Statistics /Computer Science

List of constraints

Hard Constraints

- Courses having common Student cannot assign at the same time period 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-periods.

Solution

Taking each course as a vertex, edge between two vertices is drawn only if there is common student. (See Figure-3)

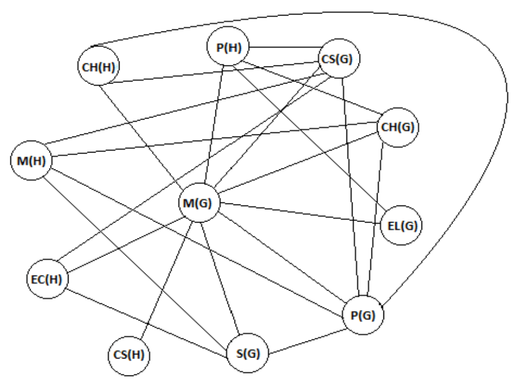

Figure 3 Course Conflict Graph

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

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

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

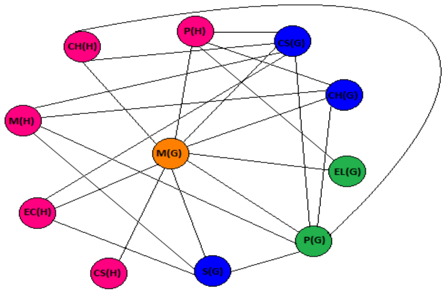

Figure 4 Colored Conflict Graph

RESULT

From the solution, we know that the hard constraints are satisfied properly. The resultant minimum number of time periods needed is 4 which cannot exceed the total available 8 periods. There are no common students in between different Honourssubjects, hence allotted in period-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 Figure-5. All of these scheduled courses are non-conflicting combinations.

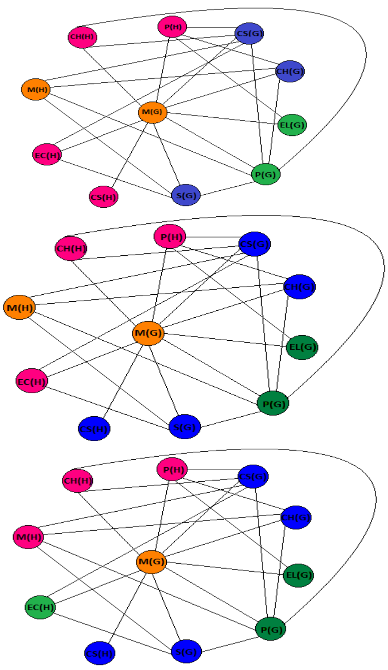

Figure 5 Alternative Solutions

Example: - 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 [22] [23] 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

Periods (P)	N_1	N_2	N_3	N_4	N_5
T_1	2	0	1	1	0
T_2	0	1	0	1	0
T_3	0	1	1	1	0
T_4	0	0	0	1	1

List of constraints

Hard Constraints

- At any period, each subject can teach by at most 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 Figure-6. The chromatic number acts as the minimum number of periods.

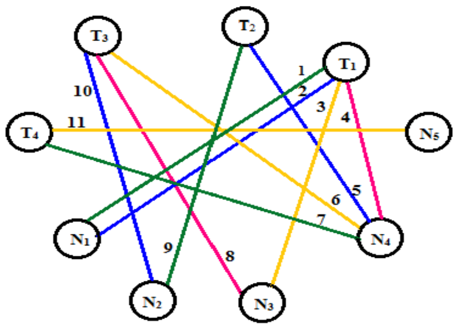

Figure 6 Bipartite Graph G

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) [9], and a proper vertex coloring of the line graph shown in Figure-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 Figure-6. acts as the vertices of L(G) in Figure-7.

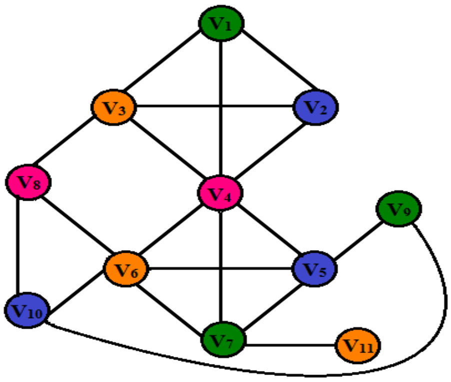

Figure 7 Line graph L(G)

RESULT

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

Table 3 Graph Coloring Solution Table of Figure-7

GREEN	BLUE	PINK	YELLOW
V_1	V_2	V_4	V_3
V_7	V_5	V_8	V_6
V_9	V_{10}		V_{11}

Now, each of these colors in Table-3 represents periods in Table-4 and the vertex in L(G) (Figure-7) that corresponds to a particular edge in the bipartite graph G (Figure-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
T_1-N_1	T_1-N_1	T_1-N_4	T_1-N_3
T_2-N_2	T_2-N_4	T_3-N_3	T_3-N_4
T_4-N_4	T_3-N_2		T_4-N_5

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 parallel.

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. Akbulut, A., 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.
2. Birkhoff, G.D. (1912). “A determinant formula for the number of ways of coloring a map,” *Annals of Mathematics,* 14, pp. 42-46.
3. Bondy, J. A. (1969). “Bounds for the chromatic number of a graph,” *Journal of Combinatorial Theory*, 7, pp. 96-98.
4. Bondy, S. (2002). “Final Examination Scheduling,” Communications of the ACM, 22(7), pp. 494-498.
5. Bresina, J. (1996). “Heuristic-Biased Stochastic Sampling,” Proceedings of the 13th National Conference on Artificial Intelligence, pp. 271-278.
6. Burke, E. K., Elliman, D. G., & 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.
7. Burke, E.K., Elliman, D.G., & 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.
8. Carter, M. W., Laporte, G., & Lee, S. Y. (1996). “Examination timetabling: Algorithmic strategies and applications,” *Journal of the Operational Research Society* 47(3), pp. 373-383.
9. Deo, N. (2014). “Graph Theory with Applications to Engineering and Computer Science”, Prentice Hall of India, Delhi, India.
10. Dobrolowski, T., Dereniowski, D., & 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.
11. Garey, R. D., Johnson, S. (1979). “Computers and intractability: A Guide to the Theory of NP-Completeness,” Freeman.
12. Heawood, P. J.(1980). “Map-Colour Theorems, 1980,” Quarterly Journal of Mathematics, Oxford, 24, pp. 332–338.
13. Johnson, D. S., Aragon, C. R., McGeoch, L. A., & Schevon, C. (1991) “Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph

Coloring and Number Partitioning," Operations Research, 39(3), pp. 378-406., The IBM Research Symposia Series, New York, NY: Plenum Press, 1972, pp. 85-103.
14. 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.
15. Kenneth, A., Wolfgang, H. (1977). "Every Planar Map is Four Colorable. I. Discharging," *Illinois Journal of Mathematics,* 21(3), pp. 429-490.
16. Kiaer, L., Yellen, J. (1992). "Weighted Graphs and University Timetabling," Computers and Operations Research, 19(1), pp. 59-67.
17. Klotz, W. (2002) "Graph Coloring Algorithms," Mathematic- Bericht 5, TU Clausthal, pp. 1-9.
18. Miner, S.K., Elmohamed, S., & Yau, H.W. (1995). "Optimizing Timetabling Solutions using Graph Coloring," NPAC, Syracuse University, pp. 99-106.
19. Mondal, B., De, K. (2017). "Overview Applications of Graph Theory in Real Field". *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2(5), pp. 751-759.
20. Peck, J. E. L., Williams, M. R.(1966). "Examination Scheduling," Communications of the ACM, 9(6), pp. 433-434.
21. Redl, T.A.(2007). "University Timetabling via Graph Coloring: An Alternative Approach," Congressus Numerantium, 187, pp. 174-186.
22. Sarma, S. V. M.(2012). "Application of graph theory in human life," *International Journal of Computer Application*, 1(2), pp. 21-30.
23. Shirinivas, S. G., Vetrivel, S., & 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.
24. Torkestani, J.A., Meybodi, M.R.(2009). "Graph Coloring Problem Based on Learning Automata," *International Conference on Information Management and Engineering*, pp. 718-722.
25. Welsh, D.J.A., Powell, M.B. (1967). "An Upper Bound for the Chromatic Number of a Graph and its Application to Timetabling Problems," *The Computer Journal.* 10(1), pp. 85-86.
26. Wood, D. C.(1968). "A System for Computing University Examination Timetables," The Computer Journal, 11, pp.41-47.
27. Wood, D. C.(1969). "A Technique for Coloring a Graph Applicable to Large Scale Timetabling Problems," *The Computer Journal,* 12, pp. 317-319.

How to cite this article:

Narayan Poddar and Basudeb Mondal.2018, An Instruction on Course Timetable Scheduling Applying Graph Coloring Approach. *Int J Recent Sci Res.* 9(2), pp. 23939-23945. DOI: http://dx.doi.org/10.24327/ijrsr.2018.0902.1567

****