International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Time-Table Scheduling Problem: A Case Study of Science College

Dr. Pratiksha M. Kadam

Assistant Professor, Kishinchand Chellaram College, HSNC University, Mumbai, Maharashtra, India

Abstract
This paper presents a design for a weekly timetable for a Science college that optimally allocates classrooms, faculty members, and time slots to various science courses using graph theory techniques. This design uses a mathematical approach to maximize utilization of resources and minimize conflicting times for different classes. The paper provides a detailed description of the techniques used, the models created, and the solutions obtained. It also discusses various considerations that need to be considered when designing a weekly timetable, including the use of software systems and algorithms to optimize the design. Paper explains the Application of Graph Theory in scheduling problem with a case study of the Science college.

Keywords: Graph, Breadth-First Search, Greedy Algorithm, nodes, edges, graph coloring

Introduction
The design of an effective and efficient weekly timetable for a Science college is a challenging task. It requires careful consideration of the resources available (classrooms, faculty, and timeslots), the courses to be offered, and possible clashes between classes. This is especially true in a modernized society where students require flexible courses and teachers have more diverse schedules. To meet these demands, it is essential to use efficient methods for optimizing the utilization of resources and avoiding conflicts between classes. (Carter, 1996)
The use of graph theory techniques is one such approach to designing an optimal timetable. Traditional graph theory techniques involve creating a graph to represent the pertinent resources and courses, and then seeking for solutions that optimize the utilization of resources and minimize conflicts between classes. In this paper, we will discuss the use of graph theory techniques for the design of an effective weekly timetable for a Science college. We will also discuss the various considerations and models involved in the design process, as well as the potential benefits of such a design. (Sallam, 2011)
The mathematical basis of graph theory revolves around the definition of “graphs” and their associated “vertices” and “edges”. A graph can be thought of as a network composed of a set of vertices (“nodes”) joined by edges (“arcs”). A typical graph consists of a set of vertices V and a set of edges E. This network of vertices and edges can be used to represent various kinds of relationships between the vertices. (M, 2007)
In the context of designing a weekly timetable for a Science college, the vertices represent the available resources (classrooms, faculty, and timeslots) and the edges represent the various courses to be offered by the college. In order to solve the problem of designing an optimal timetable, it is then necessary to assign resources to courses in order to maximize the utilization of resources while minimizing conflicts between

classes. This can be done by creating a model on the graph that assigns each course to the resources it requires and then attempting to optimize the model.

Review of Literature

- The paper "A Graph Theory Approach to University Timetabling" by Carter, M. W., Laporte, G., Lee, S. Y., & Luh, P. B. published in the journal Interfaces in 1996 is a notable contribution to the field of university timetabling using graph theory. In this paper, The authors has addressed the complex task of university timetabling, a problem that involves scheduling courses, classrooms, and faculty members while adhering to various constraints and preferences.
- The paper titled "A Survey of Graph Coloring Algorithms" by Bona, M., published in The Electronic Journal of Combinatorics in 2007, is a valuable resource that provides an overview of various graph coloring algorithms. Paper presents a comprehensive survey of graph coloring algorithms, highlighting different approaches, strategies, and their applications.
- The paper titled "Graph Theory Based Algorithm for Exam Timetabling Problem" by Sallam, A. A., & Hassan, H. M., presented at the Proceedings of the World Congress on Engineering in 2011, focuses on addressing the challenging problem of exam timetabling using graph theory.

There's a significant amount of work done on the timetable scheduling problems using graph theory at national an international level. With reference to New Education Policy, the timetable scheduling is a challenging task. There is a scope to work on this problem with reference to syllabus and interdisciplinary approach teaching-learning timetable.

Models and Techniques for Designing an Optimal Timetable

There are several models and techniques that can be used to design an optimal timetable for a Science college. Some of the most commonly used techniques are: (M, 2007)

Breadth-First Search:

This technique involves searching the graph for feasible solutions that satisfy the requirements of all courses. The search is done in a breadth-first manner, which means that it starts with the courses that have the fewest requirements and then expands outward to include necessary resources until solutions are found.

Greedy Algorithms:

Greedy algorithms are designed to find a solution to a given problem by making locally optimal choices, without considering the consequences of these choices for the overall problem. That is, these algorithms seek to maximize the immediate benefits of a decision, without considering the global impact of the decision. These algorithms are often used for solving graph problems that involve finding feasible solutions while minimizing conflicts between resources.

Dynamic Programming:

Dynamic programming is an optimization technique that involves breaking down a complicated problem into simpler subproblems, solving the subproblems, and then recombining the solutions to obtain an overall optimal solution. This technique is often used for solving large scale graph problems, as it allows for a more efficient utilization of resources.

Software Systems and Algorithms

In addition to the mathematical models and techniques discussed above, it is also important to consider the use of different software systems and algorithms for optimizing the design of a weekly timetable for a science college. The use of software systems such as timetabler can simplify the design process by providing an interface for the user to create, modify, and optimize their timetables. Additionally, software systems can also include a variety of algorithms for finding optimal solutions, such as simulated annealing, genetic algorithms, and particle swarm optimization.

We can also use python programming to write the programmes that generate the graphs for the scheduling problems using in-built functions. In this papers python programme is used to explain a case study of timetable scheduling for a science college. (Sallam, 2011)

Problem Framework

Let's Consider an example of Science College where Physics, Chemistry, Maths, Statistics, Botany and Zoology are taught and the subject combination offered is PCM, PMS and CBZ. Lectures can be conducted from 09:00am to 01:00pm. Only two classrooms are assigned to conduct these lectures.

When designing a weekly timetable for a Science college, it is important to consider a framework for implementation. This includes deciding on the correct data structures, such as graphs and tables, to represent the resources and courses, as well as the choice of software systems and algorithms to be used. Additionally, it is important to decide on the appropriate measures to evaluate the effectiveness of the timetable, such as the level of utilization of resources or the minimizing of conflicting times for classes.

In this case, we have six subjects - Physics, Chemistry, Maths, Statistics, Botany and Zoology - and three possible combinations of three subjects that cannot be scheduled at the same time. We will be using the graph edge and vertex coloring algorithm to solve this problem, and then displaying the graphical representation of the solution.

Algorithm and Solution

The first step of the graph coloring algorithm is to add edges to the graph between the vertices (subjects) that represent pairs that cannot be scheduled at the same time. This can be done by noting that the three combinations of three subjects are not compatible with each other and cannot be scheduled at the same time. We will add an edge between each pair in this set of three within a vertex.

Next, we color the graph. Each color will represent a different lecture. We will use three colors, Red, Blue, and Green, to represent the three lectures. We can then assign these colors to the vertices that represent each combination of subjects. For example, the vertex representing Physics, Chemistry, and Maths will be colored Red, the vertex representing Physics, Maths, and Statistics will be colored Blue, and the vertex representing Chemistry, Botany, and Zoology will be colored Green. (M, 2007)

Finally, we display the graph and the colors. This will provide a graphical representation of the solution.

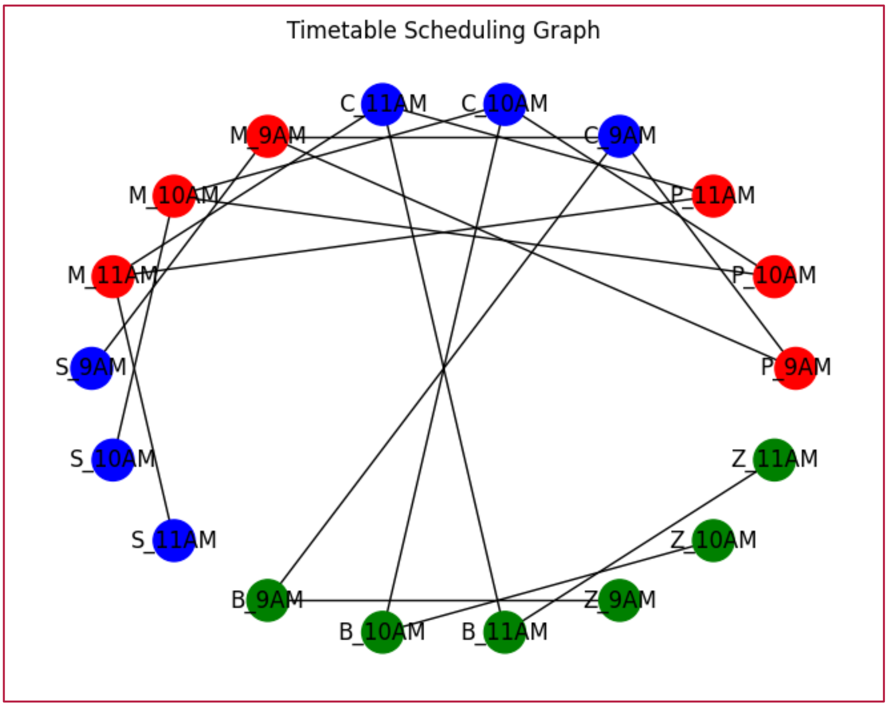
We can generate this Graph using Python Code as follows:

```
import networkx as nx
import matplotlib.pyplot as plt
def solve_timetable_scheduling(subjects, combinations, classrooms, time_slots):
# Create a graph for timetable scheduling
G = nx.Graph()
```

```
# Add nodes for subjects at different time slots
for subject in subjects:
for time_slot in time_slots:
G.add_node(subject + '_' + time_slot)
# Add edges to represent clashes (subjects in the same combination)
for combination in combinations:
for time_slot in time_slots:
for i in range(len(combination) - 1):
G.add_edge(combination[i] + '_' + time_slot, combination[i + 1] + '_' + time_slot)
return G
def color_timetable_scheduling_graph(G):
# Assign colors to vertices representing combinations of subjects
colors = {'P': 'red', 'C': 'blue', 'M': 'red', 'S': 'blue', 'B': 'green', 'Z': 'green'}

# Get colors for each node based on the subject in the combination
node_colors = [colors[node.split('_')[0]] for node in G.nodes]

return node_colors

def display_timetable_scheduling_graph(G, node_colors):
# Visualize the graph with node colors using a circular layout
pos = nx.circular_layout(G)

nx.draw(G, pos, with_labels=True, font_weight='bold', node_color=node_colors, cmap=plt.cm.rainbow,
node_size=800)

plt.title('Timetable Scheduling Graph')
plt.show()

# Problem parameters
subjects = ['P', 'C', 'M', 'S', 'B', 'Z']
combinations = [('P', 'C', 'M'), ('P', 'M', 'S'), ('C', 'B', 'Z')]
classrooms = 2
time_slots = ['9:00 AM', '10:00 AM', '11:00 AM']

# Solve the timetable scheduling graph
G = solve_timetable_scheduling(subjects, combinations, classrooms, time_slots)

# Color the timetable scheduling graph
node_colors = color_timetable_scheduling_graph(G)

# Display the timetable scheduling graph with colors
```

display_timetable_scheduling_graph(G, node_colors)

The graph generated is:



Future Scope

Interdisciplinary Timetabling

- Extend graph-based scheduling to support interdisciplinary learning experiences as encouraged by the NEP.
- Create timetables that enable students to engage in cross-disciplinary projects, workshops, or collaborative initiatives.

Assessment and Evaluation Scheduling

- Utilize graph theory to optimize the scheduling of assessments, exams, and evaluations in alignment with the continuous and comprehensive evaluation system proposed by the NEP.
- Ensure fair and efficient distribution of assessment tasks across subjects and time periods.

Flexible Academic Calendars

- Develop flexible academic calendars using graph-based approaches to accommodate variations in the duration and timing of academic terms.
- Enable academic institutions to adapt their schedules based on local needs, festivals, or community events.

Student-Centric Timetables

- Design timetables that prioritize student-centric features, such as adequate breaks, extracurricular activities, and opportunities for skill development.
- Use graph theory to balance academic and non-academic elements in the timetable.

Conclusion

In this paper, we have discussed the design of a weekly timetable for a Science college using graph theory techniques, including models, techniques, software systems, and algorithms. we have used the graph edge and vertex coloring algorithm to solve a problem involving six subjects and three combinations of three subjects that cannot be scheduled at the same time, and displayed the graphical representation of the solution. This algorithm provides an effective solution for this type of problem, as it allows for quick visualization of the possible options.

References

1. Carter, M. W. (1996). "A Graph Theory Approach to University Timetabling". *Interfaces*.
2. M, B. (2007). A Survey of Graph Coloring Algorithms. *The Electronic Journal of Combinatorics*.
3. Sallam, A. A. (2011). Graph Theory Based Algorithm for Exam Timetabling Problem. *Proceedings of the World Congress on Engineering.*