Exploring Graph Coloring Methods to Enhance Efficiency in Scheduling Issues: An Examination in the Context of Allocation and Conflict Resolution

S. Nareshkumar[1], Ponnarasu. K[2], D.Moganavalli[3], K.Kavitha[4]

[1]Department of Mathematics, Sri Krishna Adithya College of Arts And Science, Kovaipudur, Coimbatore – 641042, Email: mathsnareshkumar@gmail.com
[2] Department of Mathematics, Sri Krishna Adithya College of Arts And Science, Kovaipudur, Coimbatore – 641042, Email: ponnugoldking@gmail.com
[3] Department of Mathematics, Annai Violet Arts And Science College, Chennai – 600053, Email: moganasuba@gmail.com
[4] Department of Mathematics, Bharathi Women's College, Chennai 108, Chennai – 600053, Email: kavithak_mat@bwc.edu.in

Received: 13.04.2024 Revised : 16.05.2024 Accepted: 21.05.2024

ABSTRACT
This paper delves into scheduling challenges within graph theory, emphasizing goals of optimization, fairness, and conflict resolution. The approach involves utilizing graph coloring, where tasks are depicted as vertices and conflicts as boundaries in a chart. The application of the Welsh-Powell algorithm is explored to lessen the requisite sum of colors. Through a case study, the paper demonstrates the utilization of graph coloring to address a scheduling issue, determining the minimum workforce required while adhering to specified constraints. Additionally, the paper discusses real-world applications of graph coloring across various domains such as regional planning, wireless communication networks, scheduling, and timetabling, underscoring its adaptability in solving optimization problems.

Keywords: Graph coloring, Vertex coloring, Chromatic Number, Scheduling.

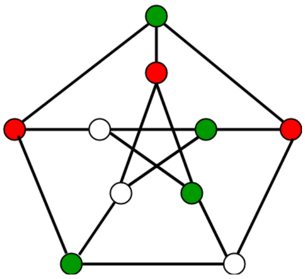
1. Preliminaries

"The foundations of graph theory trace back to Leonhard Euler's landmark paper 'Seven Bridges of Konigsberg' in 1736, where the study of networks began. The term 'graph' itself wasn't coined until Sylvester's 1878 paper, and the first comprehensive textbook emerged in 1936 by Denes Konig, later followed by Frank Harary's influential 1969 book, widely used in graph theory studies[1]. Graph theory found practical applications in various fields. Gustav Kirchhoff utilized graphs to compute electrical currents in networks, extending to the enumeration of chemical molecules. A.F. Mobius introduced complete and bipartite graphs in 1840, while Thomas Guthrie initiated the renowned four-color problem in 1852, which took a century to resolve by Kenneth Appel and Wolfgang Haken. Initially applied to planar graphs and map coloring, Heawood established the five-color theorem in 1890, stating that any planar map could be highlighted with five or fewer colors. George David Birkhoff, in 1912, pioneered the chromatic polynomial for algebraic graph theory, extending the study of coloring problems. The practical applications of graph coloring are vast, from map and timetable scheduling problems to network design, Sudoku, register allocation, and detecting bipartite graphs. Its significance lies in solving complex optimization problems, such as conflict resolution and optimal event partitioning.

2. Basic Definitions and Results

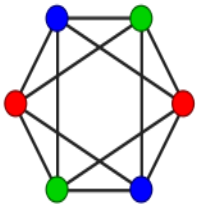
Definition 1: **Graph**

A graph G is characterised by two components: V(G), a finite set with elements referred to as vertices or points, and X(G), a set containing unordered pairs of distinct elements from V(G), known as edges or lines within the graph. (or) The ordered twosome of vertices and edges are called graph.

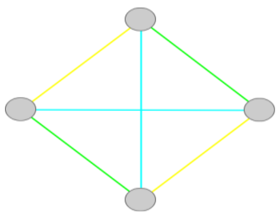


Definition 2: **Graph coloring**
Graph colouring denotes to the problem of coloring vertices of a graph in such a way that no two adjacent apexes have the same color. This is also called the vertex coloring problem. If coloring is done utilising at most m colors, it is called m-coloring

Definition 4:Vertex coloring
Vertex coloring is a method practiced to allocate colors to the vertices of a graph in such a way that **no** two adjacent vertices have the same color**.**



Definition 5:Edge coloring
Edge coloring is a technique used to assign colors to the edges of a graph in such a method that no two incident edges have the same color.



Definition 6: Chromatic Number
The chromatic number can be defined as the smallest number of colors required to appropriately color any graph. In other words, the chromatic number can be labeled as a minimum amount of colors that are desired to color any graph in such a way that no two adjacent vertices of a graph will be given the same color..
$\chi(G)$ = least number of colors needed to color a graph.

3. Some Claims of Graph Coloring in Real-World Networks
The graph coloring problems have a huge number of applications. These applications highlight the versatility of graph coloring in cracking optimization problems and improving the efficiency of various processes in different real-world network scenarios.
1. Map Coloring for Regional Planning: The first represents geographical sections on a map with diverse colors such that no two adjacent regions share the same color. Graph coloring is applied in regional planning to assign colors to geographical parts on a map. The objective is to color neighboring regions differently, representing, for example, different partitioning regulations or political boundaries.
2. Wireless Communication Networks: In wireless communication, graph coloring is castoff to assign rates to transmitters to avoid intrusion. Adjacent transmitters must use different frequencies to minimize signal overlap and maintain the quality of communication.
3. Register Allocation in Compilers: Compilers use graph coloring to assign registers to variables in a way that minimizes the number of registers required, enhancing the performance of amassed code.

4. Traffic Light Control: Graph coloring can be employed to optimize traffic light timing at intersections. The goal is to avoid conflicting movements and enhance the flow of traffic.
5. Circuit Board Design: In the design of integrated circuits, graph coloring is used to assign channels to networks, ensuring that no two adjacent connections share the same channel to stop interference.
6. Resource Allocation in Cloud Computing: Graph coloring can be used in cloud computing environments to schedule tasks on different servers, avoiding resource conflicts and optimizing task completion times.
7. VLSI Design: In Very Large Scale Integration (VLSI) design, graph dying is used for cell placement, determining the locations of various components on a chip while avoiding overlap and ensuring efficient wiring.
8. Radio Frequency Identification (RFID): In RFID systems, graph coloring helps prevent collisions between signals from different RFID tags by assigning distinct time slots or frequencies to the tags.
9. Social Network Analysis: Graph coloring techniques are applied in social network analysis to detect communities or groups of closely connected individuals.
10. Sudoku: Sudoku puzzles can be approached and solved using graph theory concepts, particularly by representing the riddle as a graph and applying graph algorithms to explain it.
11. CCTV installation: In CCTV installation can be seen as a variant of the "vertex cover" problem, which is a fundamental concept in graph theory.
12. GSM mobile phone Network: Within GSM mobile phone networks, the principles of graph theory find application across various facets, encompassing Network Topology Modeling, Coverage and Connectivity, Routing and Optimization, Quality of Service (QoS), Network Resilience, and Handover Management.
13. Scheduling and Timetabling: Assigning time slots or schedules to events or tasks to prevent clashes is a crucial task, especially in academic settings. Graph coloring serves as a method to schedule classes effectively, ensuring no overlap between classes sharing the same room or instructor. Scheduling and timetabling entail intricate processes that involve allocating resources—be it time, space, or equipment—for a set of activities while meeting diverse constraints. Utilizing graph coloring offers a robust solution framework for tackling these complex scheduling and timetabling challenges across various domains.

4. The Scheduling Problem

Graph coloring provides solutions to many scheduling problems. In detail, a set of jobs need to be appointed to specific time slots. They may be scheduled in any sort. However, the problem is that jobs can conflict, meaning they cannot be allocated to the same time slot as they might be sharing the same resources. The graph consists of vertices for each job and edges for each couple of conflicting jobs. The scheduling problems in graph theory, the objective is to discover the most effectual way to plan a set of jobs on machines or workers. There are three main objective of arranging problems which are

1. Optimization issues
2. Equity and
3. Conflict resolution:

1. The Optimization is meant to make the most ofthe profit and minimize the cost, for example, scheduling machine time for the earliest accomplishment time.
2. Equity implied making things reasonable to all the participants, for instance, scheduling baseball games by making a precise number of home and away games.
3. Conflict resolution avoids conflicts, such as scheduling college final examinations for the end term.

Graph coloringis a problem that designated colors in the graph for limits. Vertices of the graph will be colored so that no two adjacent vertices take the identical color. It does not matter which one to select first. In the algorithm [4], there are steps that we need to track while coloring the vertices.
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.
Step 1: Input the conflict graph G.
Step 2: Calculate degree sequence of the input conflict graph G.
Step 3: Allocate color1 to the vertex vi of G having maximum degree.
Step 4: Allot color1 to all the non-adjacent uncolored vertices of vi and store color1 into Used Color array.
Step 5: Assign novel color which is not formerly used to the next uncolored vertex having next highest degree.
Step 6: Assign the similar new color to all non-adjacent uncolored vertices of the freshly colored vertex.
Step 7: Reiterate step-5 and step-6 until all vertices are colored.

Step 8: Set the least number of non-conflicting time-slots n= chromatic number of the colored graph=total number of elements in Used Color array.
Step 9: End
An algorithm[8] named Welsh-Powell is the one that guarantees the lowest number of colors in a graph, and the algorithm is as follows:

- Select the first vertex named A having the most incoming edges and color it.
- Look at other vertices and colour if
  1) They are not neighbors
  2) Vertices are undyed
  3) The vertex to which it is adjacent is not colored with the same.
- Repeat the second step until you no more vertices to color.

The below example is of a scheduling problem solved with graph coloring to find optimal solution.

Problem 1
Suppose there are 6 jobs that need to be assigned to workers. Each worker can take any jobs, and some jobs cannot be assigned to the same worker due to some conflict, which is given below

Jobs	A	B	C	D	E	F
Constraints / Conflict	B,C	A,C,E	A,B,E,D	C,F	B,C,F	D,E

Find the minimum sum of workers (optimal solution) needs to assign the jobs.

Solution
To solve this problem by using graph coloring,
Let us draw asix points A, B, C, D, E and F which is representing each type of jobs and then join the vertices according to the constrains/conflict.

Graph representation
Vertices(Job): represents the jobs
Edges(Constrains): represents that the corresponding jobs cannot be assigned to the same worker, and
Colors: represents the workers to be assigned.
We represent the constrains as an undirected graph, where each jobs is a node (or) vertices and edge between the nodes indicates that the corresponding jobs can't be assigned to the same worker.
Graph Construction

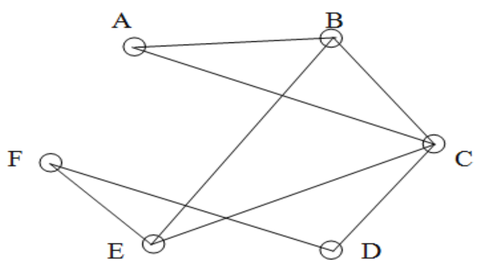


Figure-1

Color Assignment

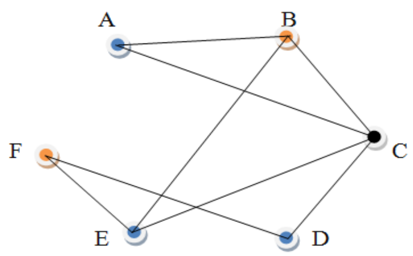


Figure-2

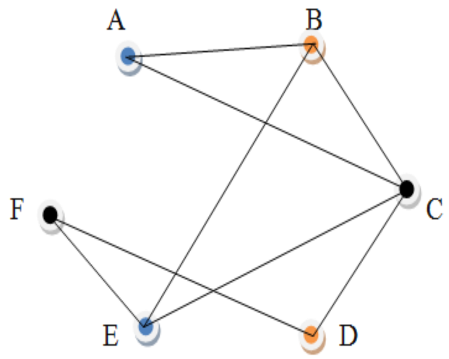


Figure-3

One possible way to assign the colors (workers) to nodes (jobs) in figure-2.This graph suggests a minimum of 3 colors (workers) are needed.

Worker 1 (Blue)	Worker 2 (Red)	Worker 3 (Black)
Job A, D& E	**Job B &F**	**Job C**

Another possible way to assign the colors (workers) to nodes (jobs) in figure-3.This graph suggests a minimum of 3 colors (workers) are needed.

Worker 1 (Blue)	Worker 2 (Red)	Worker 3 (Black)
Job A & E	**Job B & D**	**Job C & F**

Each worker will be assigned a set of jobs that can coexist based on the constraints provided.
According to the main objective of scheduling problems (Optimization issues, Equity and Conflict resolution) figure-3 is the optimal scheduling. By analyzing the graph and constraints, we've determined that 3 workers are needed to complete all the jobs without violating the constraints.
The problem of assigning 6 jobs to workers while adhering to specified constraints was approached using graph coloring techniques. By representing the jobs and their conflicts as nodes and edges in an undirected graph, respectively, it was determined that a minimum of 3 workers are needed to complete all the jobs without violating the constraints
The graph coloring algorithm aided in assigning colors (representing workers) to the nodes (jobs) in such a way that no adjacent nodes (jobs that cannot be together) have the matching color. Therefore the problem was positively solved by determining the minimum number of workers needed and assigning jobs to them based on the restraints, showcasing the efficacy of graph coloring in resolving scheduling predicaments with differingsupplies.

Problem 2
The time table shows with an X pair of course whichhave one or more students in common, only 2 class room available for use at any time. To enterprise an competent way to schedule the examination without any conflict.

	French	Maths	History	Philosophy	English	Italian	Spanis	Chemistry
French	-	X	-	X	X	-	-	X
Maths	X	-	-	-	-	X	X	-
History	-	-	-	X	-	-	X	-
Philosophy	X	-	X	-	X	-	X	-

English	X	-	X	-	-	-	-	X
Italian	-	X	-	-	X	-	-	X
Spanis	-	X	X	X	X	-	-	-
Chemistry	X	X	-	-	X	X	-	-

Solution

To design an efficient schedule for the examination without conflicts, we can use a graph coloring approach. Each course can be represented as a vertex, and an edge exists amid two vertices if the equivalent courses have one or more students in common (i.e., these courses cannot be scheduled at the same time).

From the given data, create a graph where each course is a vertex and an edge occurs between two vertices if they share common students. Using the above matrix here's the adjacency matrix representation is

- French – Maths (F-M)
- French – Philosophy (F-P)
- French – English (F-E)
- French – Chemistry (F-C)
- Maths – Italian (M-I)
- Maths – Spanish (M-S)
- History – Philosophy (H-P)
- History – Spanish (H-S)
- Philosophy – Spanish (P-S)
- English – Chemistry (E-C)
- English – Philosophy (E-P)
- Italian – Chemistry (I-C)
- Spanish- English (S-E)
- Chemistry-Maths (C-M)

Graph Construction

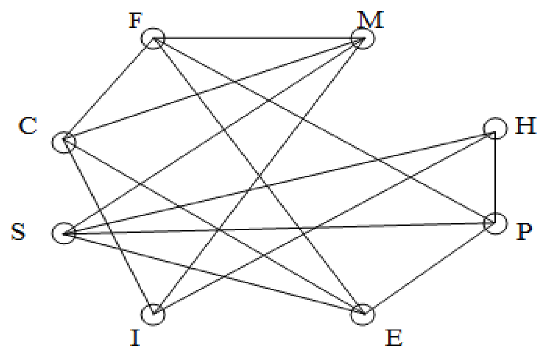


Figure-1

Graph Coloring

Using the Group Coloring algorithm to assign time slots:

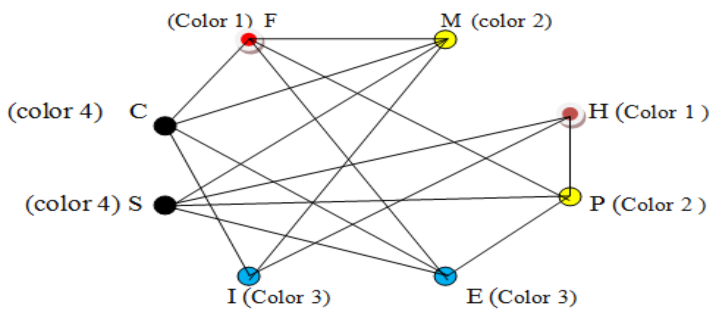


Figure-2

1. French - Color 1
2. Maths - Color 2
3. History - Color 1
4. Philosophy - Color 2
5. English - Color 3
6. Italian - Color 3
7. Spanish - Color 4
8. Chemistry - Color 4

Assign Time Slots and Classrooms
Since there are only two classrooms available, we need to ensure no more than two exams are scheduled at the same time.

Final Examination Schedule with Time Slots

- **Time Slot 1:**
  - Classroom 1: French
  - Classroom 2: History
- **Time Slot 2:**
  - Classroom 1: Maths
  - Classroom 2: Philosophy
- **Time Slot 3:**
  - Classroom 1: English
  - Classroom 2: Italian
- **Time Slot 4:**
  - Classroom 1: Spanish
  - Classroom 2: Chemistry

Conclusion
The above schedule ensures that no two exams with common students are scheduled simultaneously and that only two classrooms are used at any given time.

CONCLUSION

Various scheduling scenarios, including exam scheduling, employee shift scheduling, conference scheduling, and more, are examined to illustrate the versatility of graph coloring techniques in optimizing scheduling processes across diverse domains. Additionally, the paper presents a problem-solving methodology utilizing graph coloring, showcasing how constraints can be represented and efficiently solved using graph coloring algorithms to assign workers to jobs. Moreover, the paper delves into real-world applications of graph coloring in networks such as regional planning, wireless communication, compiler optimization, traffic light control, and many others, demonstrating the wide range of problems that can be effectively addressed using graph coloring techniques. In conclusion, the paper emphasizes the critical role of graph coloring in tackling complicated scheduling and optimization encounters across various fields, accentuating its practical significance in enhancing effectiveness and resolving skirmishes in practical circumstances.

REFERENCES

[1] Najera, J. (2018, November 27). Graph Theory — History & Overview. Towards Data Science. Available: https://towardsdatascience.com/graph-theory-history-overview-f89a3efc0478

[2] Birkhoff, G. D. (1912). A determinant formula for the number of ways of coloring a map. Ann. of Math., 14, 42-46

[3] Descartes, B. (1954). Solution to advanced problem no. 4526. Arner. Math. Monthly 61, 352

[4] A Study on Course Timetable Scheduling using Graph Coloring Approach,- Runa Ganguli and Siddhartha Roy. https://www.ripublication.com/ijcam17/ijcamv12n2_26.

[5] Wikipedia contributors. (2021b, August 5). Graph theory. Wikipedia. https://en.wikipedia.org/wiki/Graph_theory

[6] Lovasz, L. (1968). On chromatic number of finite set systems. Acta Math. Acad. Sci. Hungar., 19, 59-67

[7] Robin J. Wilson. (2013, December 17). History of Graph Theory from: Handbook of Graph Theory CRC Press. Available: https://www.routledgehandbooks.com/doi/10.1201/b16132-3

[8] Welsh, A., & Powell, B. (1967). An upper bound for the chromatic number of a graph and its application to timetabling problems. The Computer Journal,10(1), 85-86.
[9] Wikipedia contributors. (2021d, August 20). Complete graph.
[10] Wikipedia. https://en.wikipedia.org/wiki/Complete_graph
[11] Gross, J., & Yellen, J. (1999). Graph Theory and Its Applications. CRC Press, 13.
[12] Trudeau, R. J. (1994, February 9). Introduction to Graph Theory. Dover Publications.
[13] Deo, N. (1990). Graph theory with applications to engineering and computer Science. Prentice Hall of India.
[14] Dutton, R., & Brigham, R. (1981). A new graph coloring algorithm. Computer Journal, 24(1), 85–86.