International Journal of Engineering Science Invention Research & Development; Vol. I Issue VIII February 2015
www.ijesird.com e-ISSN: 2349-6185

SOME THOUGHTS ON COLOURING OF GRAPHS FOR COMPUTER APPLICATIONS

P. Arumugam
Research Scholar, Department of Mathematics, Osmania University, Hydrabad-500 007
Email: rmgam4@gmail.com

Abstract-**Data processing from data mining, image segmentation, clustering, image capturing are the areas where large data have to be distinguished for understanding and interpretation. If the graphs are colored and suitably indexed it will be highly helpful not only for the experts who are handling these graphs but will be highly useful for easy understanding for the general public. An attempt has been made in this communication to get the basics of coloring the graphs from graph theory to computer science researchers in particular and to the scientists in general.**
Keywords-**Data mining, image segmentation, data processing, coloring, graph theory**

1. INTRODUCTION

Generally graphical representations are best utilized by computer scientists in particular. The areas where they use are data processing from mining data, image segmentation, clustering, image capturing, networks etc., In these the concept of graph coloring is used in resource allocation, scheduling. Chemists will be interested in studying molecules, construction of bonds and representation of crystal structures wherein they need coloring of different atoms and molecules. Graph theory plays an important role in showing the existence of species and they have to represent migration path or movement between the regions which are located as vertices. This information is crucial to indicate the spread of disease, parasites and to study the impact of migration that affects the other species during invasion. Coloring of graphs are also important for Operation research scientists to represent the travelling salesman problem, to get optimal match of jobs, in modeling transport networks etc., Coloring of graphs embedded in surfaces will be an interesting activity for researchers involved in graph theory. The coloring problem that one normally interested in these types will have to be closely related to the notion of cycling coloring. The present aim of this communication is to briefly discuss the methods and procedures involved in coloring in general and cycling coloring in particular. [1] A graph is an assembly of .0 dots and edges as lines joining between two vertices. A directed graph is a graph where the arrows are used to assign directions to all the edges. The degree of a vertex is the number of edges attached to that vertex.].

Graph coloring techniques: Graphs are colored and this will be done as per requirements. In scheduling the methodologies may be pre coloring, list coloring, multi coloring, minimum sum coloring. In job scheduling jobs are indicated as vertices. If the jobs cannot be executed simultaneously they will be indicated as edges. Thus the graphs will clearly indicate the feasible scheduling of the jobs [1-3]. In the report [1] the aircraft scheduling has been dealt with nicely and the coloring of the vertices will be done to the flights which overlap and the two vertices will be connected if the corresponding time intervals overlap. In the report [4-6] bi processor tasks modeling are also provided by graph theory. Here vertices are earmarked for processes. If there is any task that has to be executed on processors I an d j then an edge to be added between the two vertices and edges will be colored in such a way that every color appears at most once at a vertex. No two tasks require the same two processor can be easily understood from the absence of multiple edges in the graph. In the same report [1, 7, 9] for a job to be processed only by certain machines and only in certain time slots modeling will be done through list coloring.

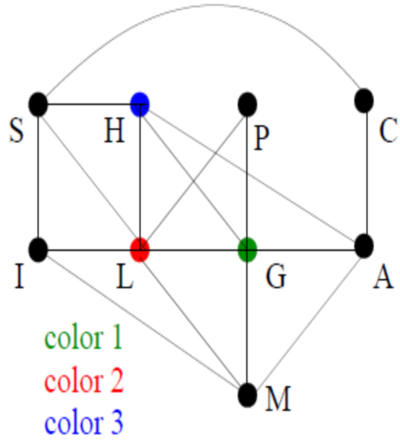
A proper coloring is an assignment of colors to the vertices of a graph so that no two adjacent vertices have the same 1 color. A k-coloring of a graph is a proper coloring involving a total of k colors. A graph that has a k-coloring is said to be k-colorable. So what we seek is a k-coloring of our graph with k as small as possible. Here is a 4-coloring of the graph is illustrated [8]

One cannot get this with less than three colors, since G, H, and L are all adjacent to each other. We shall suppose that we can properly color the graph with only three colors, and show that this leads to a contradiction. We start by coloring G, H, and L three different colors, as we must. [The chromatic number of a graph is the minimum number of colors in a proper coloring of that graph.]. The procedure requires us to number consecutively the colors that we use, so each time we introduce a new color, we number it also. Here is the procedure:

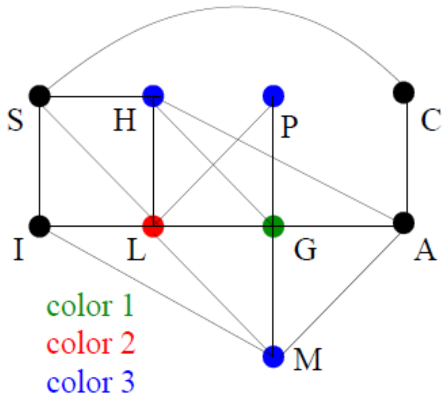
1. Color a vertex with color 1.
2. Pick an uncolored vertex v. Color it with the lowest-numbered color that has not been used on any previously-colored vertices adjacent to v. (If all previously-used colors appear on vertices adjacent to v, this means that we must introduce a new color and number it.)
3. Repeat the previous step until all vertices are colored.

Clearly, this produces a proper coloring, since we are careful to avoid conflicts each time we color a new vertex. How many colors will be used? It is hard to say in advance, and it depends on what order we choose to color the vertices.

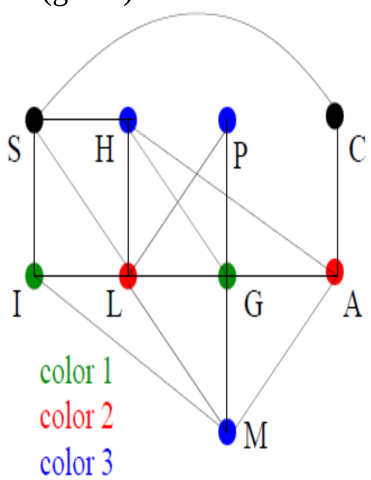
For example, suppose we decide to color the course conflict graph using the Greedy Coloring Algorithm, and we decide to color the vertices in order G, L, H, P, M, A, I, S, C. Then we would color G with color 1 (green), L with color 2 (red) since adjacency with G prevents it from receiving color 1 (green), and we color H with color 3 (blue) since adjacency with G and L prevents it from receiving colors 1 and 2 (green and red). So we have



P and M also cannot receive colors 1 and 2 (green and red), so they are given color 3 (blue):

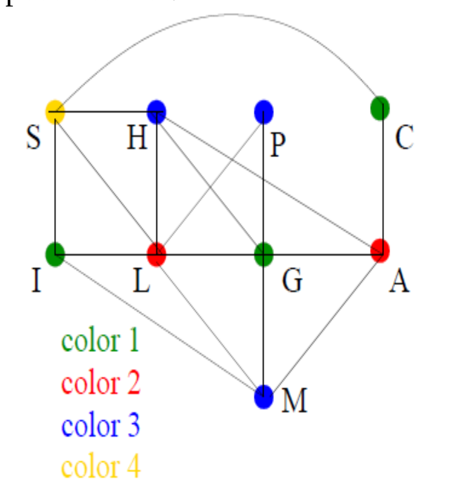


Then A cannot receive colors 1 and 3 (green and blue), so we give it color 2 (red), while I, cannot receive colors 2 and 3 (red and blue), so we give it color 1 (green).

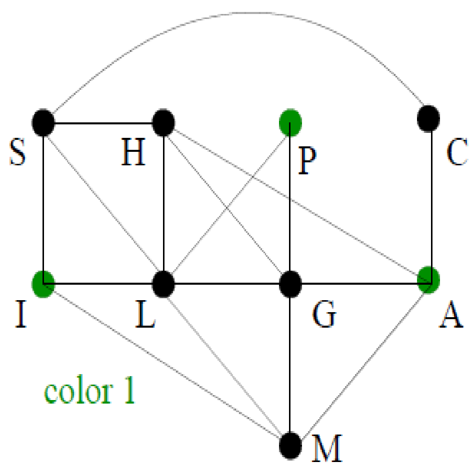


color 1

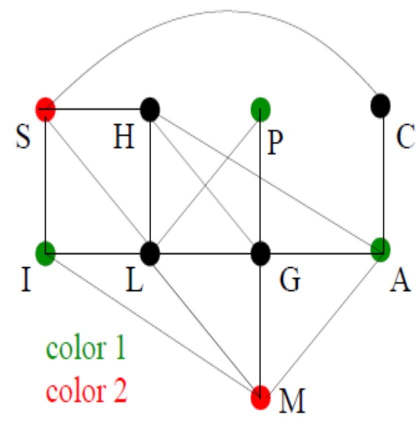
Vertex S cannot receive color 1, 2, or 3, and so we give it color 4 (say, yellow). Vertex C cannot receive color 2 or 4 (red or yellow), so we give it color 1 (green). We obtain the same coloring we had proposed earlier:



On the other hand, we could imagine choosing a deferent order. Suppose we chose to color the vertices in the order A, I, P, M, S, C, H, L, G. First we color A with color 1 (green) and also I and P color 1.

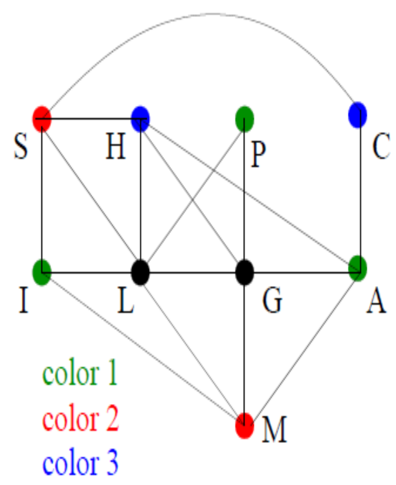


Then we color M and S with color 2 (red), because they cannot be colored green.



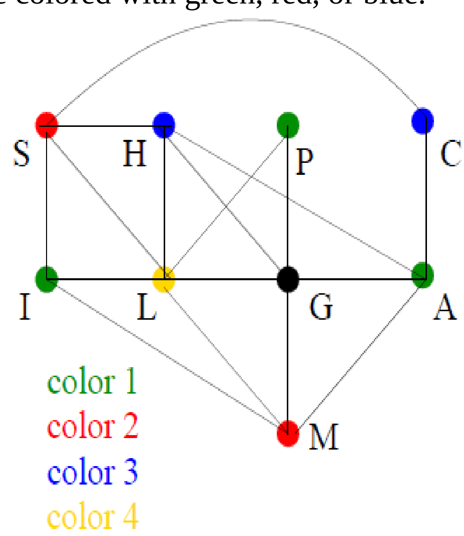
Then we color C and H with color 3 (blue), because they cannot be colored green or red

.

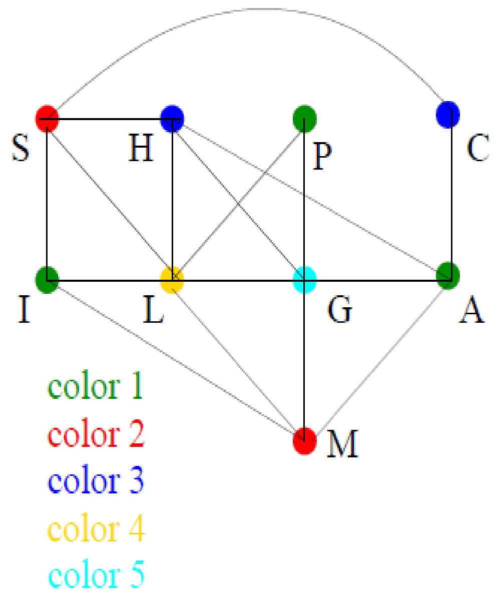


color 1

We color L with color 4 (yellow), because it cannot be colored with green, red, or blue.



We color G with color 5 (light blue), because it cannot be colored with any of the four other colors.



Algorithms:
Some algorithms are as follows:
1. Shortest path algorithm in a network
2. Finding a minimum spanning tree
3. Finding graph planarity
4. Algorithms to find adjacency matrices.
5. Algorithms to find the connectedness
6. Algorithms to find the cycles in a graph
7. Algorithms for searching an element in a data structure (DFS, BFS) and so on.

Research is being carried out in the above mentioned areas where lots of algorithms have been developed [10]

Computer languages that are developed to support graph theory in coloring graphs:

Various computer languages are used to support the graph theory concepts. The main goal of such languages is to enable the user to formulate operations on graphs in a compact and natural manner

Some graph theoretic languages are
1. SPANTREE–To find a spanning tree in the given graph.
2. GTPL – Graph Theoretic Language
3. GASP – Graph Algorithm Software Package
4. HINT – Extension of LISP
5. GRASPE – Extension of LISP
6. IGTS – Extension of FORTRAN
7. GEA – Graphic Extended ALGOL (Extension of ALGOL)
8. AMBIT – To manipulate digraphs
9. GIRL – Graph Information Retrieval Language
10. FGRAAL – FORTRAN Extended Graph Algorithmic Language [7]

Conclusions

Through this communication brief account of coloring of graphs has been provided which will be highly useful in computer science in particular and other disciplines in general. This will help in data processing and data interpretation. This will help the Radiologists and technicians to process the data further in clinical testing's.

REFERENCES

[1] Shrinivas, G. S., Vetrivel. S and Elango, "Applications of Graph theory in computer science on overview". Int. J. Eng. Sci. Techno. 2010. 2(9), 4610-4621.
[2] http://www.britinaca.com/bps/additionalcontent/18/33373769/concepts-of-graph-theory-relevant-to-Adhoc-Networks
[3] http://www.dharwadker.org/vertex_cover/
[4] http://www.icaen.uiowa.edu/~dip/LECTURE/Understanding6.html
[5] wj:math.sfsu.edu/beck/teach
[6] wikipedia.org/wiki/Median graph
[7] http://en.wikipedia.org/wiki/Bipartite_graph
[8] https://www.google.co.in/?gws_rd=ssl#q=Non-rainbow+colorings+of+3-%2C+4-+and+5-connected+plane+graphs
[9] http://en.wikipedia.org/wiki/K-means_clustering
[10] http://en.wikipedia.org/wiki/Geometric_spannerWiki