Optimized Approach for Graph Coloring

Kodathala Sai Varun[1], Kandagadla Ashok Kumar[2]

[1,2] Department of Electronics and Communication Engineering,
GITAM School of Technology, GITAM Deemed to be University
Bengaluru, Karnataka.
Email: [1]varunrail1111@gmail.com, [2]kandagadla.ashokkumar@gmail.com.

Abstract- Numerous practical applications can suitably prototyped with the help of a diagram where a set of points are connected using lines and those are solved to evaluate parameters effecting the algorithm. The applications include frequency distribution among antennas, concept of frequency reuse in cellular communication, printed circuit boards, scheduling. The problem statement of graph coloring is NP-Hard and no polynomial or exponential algorithm is provided for its solution. In this paper we provided the new algorithm for graph coloring to solve vertexes and to find chromatic number. The algorithm is found to be simple and programmable with less computation time. The mathematical analysis is done using MATLAB software ang graphic user interface is developed for responsive nature. The comparison with the former algorithms is provided and concluded the merits of the proposed algorithm.

Index Terms- Graph coloring, chromatic number, time complexity, mathematical abstraction

I. INTRODUCTION

Graph coloring is a process of tagging an attribute to vertices, edges and regions or planes. The main concept behind the graph theory is no two adjacent vertices or edges or planes can be assigned with same attribute [1]. The attribute is assumed to be color for further analysis and the minimum number of colors required to color the structure is known as chromatic number. It is required to develop an algorithm which minimizes the chromatic number at less computations and time complexity. The main applications like scheduling, networking and frequency allotment made many researchers and scholars to study and develop the most reliable algorithms. If u, v are two adjacent edges then $A(u) \neq A(v)$, where function $A(X)$ corresponds to value of attribute or color. The graph coloring problem is NP – hard to solve, where a definite algorithm is not designed to get optimum colors in less computations [2].

Many researchers proposed graph coloring algorithms such as Greedy based Graph Colouring, Welsh Powell [3], Graph Colouring using Backtracking, the largest saturation degree (DSATUR) [4], the Recursive Largest First (RLF) [5]. However, some of the algorithms involves high time complexity and space complexity, hence they lack in effective implementation. This article hence discuss new algorithm that computes in less time and less space with high data handling capacity.

The major application of graph colouring – the frequency allotment is utilized as the illustration to discuss the proposed algorithm and presented the results and its characteristics. The frequency allotment is very important in mobile communications and hence proper clustering is required to ensure zero interference and overlap of signals. The base station has to allot the frequency to band to all its subscribers and should ensure there is no same frequency allocated for more than one subscribers. And the same frequency can be used by other base station to their subscribers, therefore in such cases graph colouring can be applied to get extensive results.

Along with frequency allotment there are many other major applications like data mining [6], image segmentation [7], networking [8], clustering [9], register allocation [10], time tabling [11], etc. The proposed algorithm is expected to applied to all the applications mentioned.

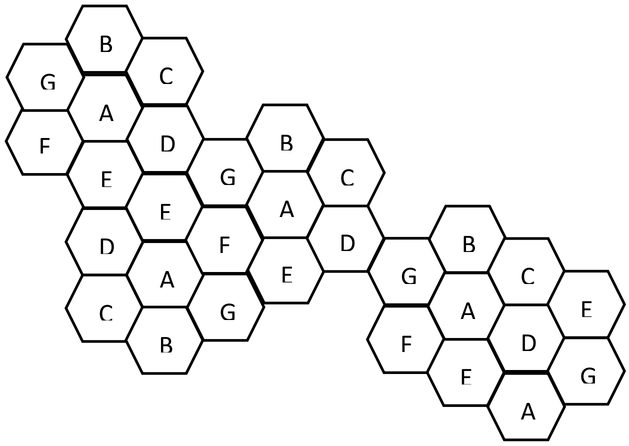

Figure 1: Concept of frequency reuse

II. Proposed Method

The proposed method can be explained using the problem statements where the vertices has to be colored with minimum number of colors. The proposed method then is applied to frequency reuse concept to illustrate the practical utilization of model.

Problem Statement – 1:

Considering the figure – 2:

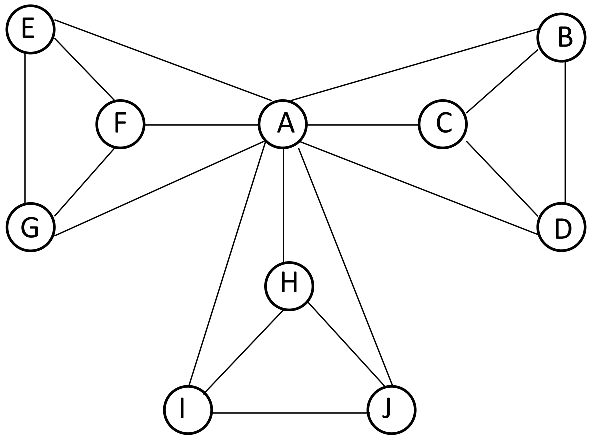

Figure 2: Problem Statement – 1

The proposed model work on basic algorithm as follows, it consists of two attributes one is color allotted and other is sign variable. The sign variable is default zero and it is complemented if color is allotted for corresponding vertex. The color is allotted on basis of first zero element in sign variable and the process is continued till all the vertices are colored. It should be noted that sign bit is reset after every iteration. Now, consider the problem statement -1 where there are 10 vertices and they are inter linked as shown in figure 2.

Graph Coloring:

Initial State:

Vertex	A	B	C	D	E	F	G	H	I	J
Color	0	0	0	0	0	0	0	0	0	0
Sign	0	0	0	0	0	0	0	0	0	0

Table 1: Initial State of attributes

Considering the vertex A, the vertex A is linked to vertices C, B, D, H, I, J, E, F, G. Since the vertices are not colored the flags are remained zero's. Therefore the first zero element is observed at first position and hence the color label 1 is allotted to A vertex after iteration 1.

The attributes are modified as follows:

Vertex	A	B	C	D	E	F	G	H	I	J
Color	1	0	0	0	0	0	0	0	0	0
Sign	0	0	0	0	0	0	0	0	0	0

Table 2: Attributes after 1st Iteration

Considering the vertex B, the vertex B is linked to vertices C, D, and A. Since the vertex A is colored the sign bit of vertex A is complemented to 1 and hence the first zero sign bit is observed at second position. Therefore the vertex B is colored with color label 2 after iteration 2.

The attributes are modified as follows:

Vertex	A	B	C	D	E	F	G	H	I	J
Color	1	2	0	0	0	0	0	0	0	0
Sign	0/1	0	0	0	0	0	0	0	0	0

Table 3: Attributes after 2nd Iteration

Similarly the process is continued for vertices C and D, but considering vertex E which has links to F, G and A. Since the vertex A is colored the sign bit is altered, the first zero element is observed at second position therefore, the vertex E is colored with similar color that B is used (i.e., 2).

The attributes are mentioned as follows:

Vertex	A	B	C	D	E	F	G	H	I	J
Color	1	2	3	4	2	0	0	0	0	0
Sign	0/1	0	0	0	0	0	0	0	0	0

Table 4: Attributes after 5 iterations

Considering the vertex F, the vertex F is linked to E, G and A. Since the vertices E and A are colored there sign bits are changed to 1, the first zero element is present at second position but E is also colored with the same color and therefore second zero element is considered which is third position therefore F is colored with same color that C is used (i.e., 3).

The attributes are mentioned as follows:

Vertex	A	B	C	D	E	F	G	H	I	J
Color	1	2	3	4	2	3	0	0	0	0
Sign	0/1	0	0	0	0/1	0	0	0	0	0

Table 5: Attributes after 6 iterations

Similarly the process is continued till all the vertices are colored, the final color allotment is given as follows:

Vertex	A	B	C	D	E	F	G	H	I	J
Color	1	2	3	4	2	3	4	2	3	4
Sign	0	0	0	0	0	0	0	0	0	0

Table 6: Attributes after final iteration

The chromatic number is given as maximum value of color that is assigned, in this case it is 4.

III. Practical Implementation

Consider the concept of frequency reuse and let the base station model is given by figure 3.

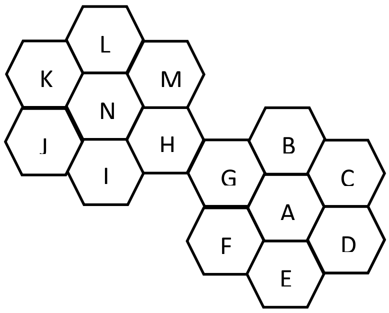

Figure 3: Base Station Model

The schematic of base station model is given by figure 4.

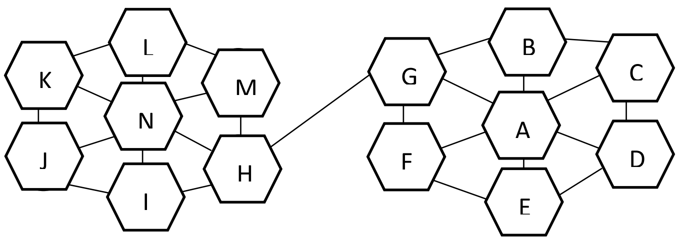

Figure 4: Schematic Model

After optimizing the base station model can be reconfigured as shown in figure 5.

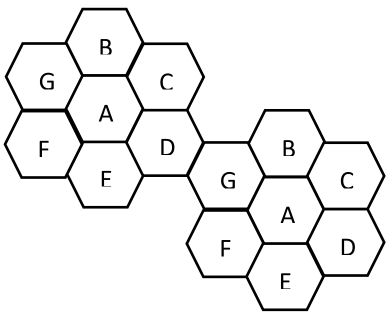

Figure 5: Optimized model

IV. ALGORITHM

1. *// nodes is the number of vertices*
2. *// link[nodes:nodes] matrix represent the connections between the vertices*
3. *// color [1:nodes] represent the colour matrix*
4. *//colasg [1:nodes] vector represent*
5. *//flag[1:nodes] used as reset matrix*
6. *//max represent the number of colours used*
7. *//final [1:nodes] represents colour assigned to vertices*
8. *val=1; max=0;*
9. *for i:1 to nodes do*

```
for j:1 to nodes  do
  flag(j)=0; assign
    for k: 1 to nodes do
      if(link(i, k) == 1 && colasg (k) > 0)  then
      flag(k)=1;
      end
      if(colasg (k) == 0) then
        colasg(val)=k;
      end
      for m: 1 to nodes  do
        if(link(i, m) == 1) then
          if(colasg(val) == colasg(m)) then
            colasg(val)=colasg(m)+1;
          end
          if(colasg(val)>max) then
            colasg(val)=max+1;
          end
        end
      end
      val=val+1;
      break
    end
  end
  max = colasg(val-1);
  final(i) = color(max);
end
```

V. Conclusions

This paper presented the optimized algorithm to compute the graph theory according to the input links corresponding to the vertices associated with other vertices. The proposed algorithm follows the time complexity of $O(n^3)$ which is polynomial in nature. The practical implementation of concept of frequency reuse is established using the proposed algorithm for graph theory and other practical applications are listed. We tested the ability of this algorithm for different graphs and appeared to be correct for all.

VI. References

1. G. L. Prajapati, A. Mittal and N. Bhardwaj, "An efficient colouring of graphs using less number of colours," *2012 World Congress on Information and Communication Technologies*, Trivandrum, 2012, pp. 666-669. doi: 10.1109/WICT.2012.6409159
2. Katerina Asdre, Kyriaki Ioannidou, Stavros D. Nikolopoulos, The harmonious coloring problem is NP-complete for interval and permutation graphs, Discrete Applied Mathematics, Volume 155, Issue 17; 2007.
3. D. J. A. Welsh, M. B. Powell; An upper bound for the chromatic number of a graph and its application to timetabling problems, The Computer Journal, Volume 10, Issue 1, 1 January 1967.
4. Brelaz D. New methods to color the vertices of a graph. Communications of the ACM,1979;22(4): 251–256.
5. Leighton FT. A graph coloring algorithm for large scheduling problems. Journal of Research of the National Bureau of Standards 1979; 84(6):489–506.
6. Smith-Miles K., Wreford B., Lopes L., Insani N. (2013) Predicting Metaheuristic Performance on Graph Coloring Problems Using Data Mining. In: Talbi EG. (eds) Hybrid Metaheuristics. Studies in Computational Intelligence, vol 434. Springer, Berlin, Heidelberg.
7. D. Gómez, J. Montero, J. Yáñez, C. Poidomani, A graph coloring approach for image segmentation, Omega, Volume 35, Issue 2; 2007

8. Awad Hassan, Maaly; Chickadel, Andrew, A Review of Interference Reduction in Wireless Networks Using Graph Coloring Methods, International journal on applications of graph theory in wireless ad hoc networks and sensor networks (GRAPH-HOC) Vol. 3, No. 1, pp. 58-67; March 2011.
9. Pierre Hansen & Michel Delattre (2012) Complete-Link Cluster Analysis by Graph Coloring, Journal of the American Statistical Association, 73:362, 397-403.
10. Gregory J. Chaitin, Marc A. Auslander, Ashok K. Chandra, John Cocke, Martin E. Hopkins, Peter W. Markstein, Register allocation via coloring, Computer Languages, Volume 6, Issue 1, 1981, Pages 47-57, ISSN 0096-0551.
11. D. J. A. Welsh, M. B. Powell; An upper bound for the chromatic number of a graph and its application to timetabling problems, The Computer Journal, Volume 10, Issue 1, 1 January 1967.