

# A Novel Approach to Solve Graph Colouring Problem

Ajay Narayan Shukla  
Department of CSE  
DIT University  
Dehradun

M. L. Garg  
Department of CSE  
DIT University  
Dehradun

## ABSTRACT

Given an undirected graph  $G = (V, E)$ , the graph coloring problem consist in assigning a color to each vertex in such a manner that two adjacent vertex have assigned different colors. The processes of assigning the colors in the graph will done in a manner such that that the total number of different colors used is minimum. Most of the existing algorithms generally deal this problem by taking consideration above constraint during assigning the color to vertices in the graph, but some time above color assignment constraints creates some other implicit constraints which increases the complexity of the algorithms. In this paper we propose an algorithm for graph coloring problem which assign the colors to vertices of the graph with minimum number of colors and without creating any other additional constraints during color assignment that required to be handled explicitly.

## General Terms

Algorithm for vertex coloring in graph.

## Keywords

Sparse Graph, Dense Graph, Register allocation

## 1. INTRODUCTION

Graph coloring is one of the most studied optimization problems in graph theory. Given a simple graph, it consists in assigning one color to each vertex such that two vertices linked by an edge get different colors and the total number of colors used is minimized.

Given an undirected graph  $G = (V, E)$ , where  $V$  is a set of vertices and  $E$  a set of unordered pairs of vertices called edges  $\{(v_i, v_j) : v_i, v_j \in V\}$  the Graph Coloring Problem (GCP) consists in assigning a color to each vertex of the graph  $G$  in such a way that any two adjacent vertices are assigned different colors, and the number of different colors used is minimized. The minimum number of colors needed to color  $G$  is called the chromatic number of  $G$ , denoted by  $\chi(G)$ . Deciding whether a graph can be colored using at most  $k$  colors is referred as  $k$ -colorability problem, which is the decision version of GCP.

Most state of the art algorithms generally deal Graph coloring problem with the explicit constraints in GCP: any two adjacent vertices should be assigned different colors, but do not specially deal with the implicit constraints between non-adjacent vertices implied by the explicit constraints. In the proposed research work this aspect of the problem shall be examined and effort shall be made to propose some new mechanism to solve the problem that must be able to work with all kinds of graph.

## 2. LITERATURE REVIEW

GCP is one of the most studied NP-hard combinatorial optimization problems. It is interesting not only for studying the computational complexity but also for practical applications. Many real world problems can be naturally

encoded into GCP and solved using a GCP algorithm. Examples of such problems include scheduling [1,2], timetable [3,4], register allocation [5], frequency assignment [6], communication networks [7] and Sudoku problem solving [15].

Algorithms for GCP generally fall into two categories: exact algorithms [8–12] and approximate algorithms. Approximate algorithms include construction method [13], local search [14]. Because of its NP-hardness, most algorithms for GCP in the literature are approximate and few are exact. Indeed, a GCP encoding a real application problem is often very large and can be out of the reach of exact algorithms, because exact coloring algorithms generally need to enumerate the search space of a problem and have difficulties in solving large instances. However, approximate algorithms are usually unable to prove the optimality of their solutions. Especially, they are usually unable to prove the unfeasibility of a  $k$ -colorability problem. So exact algorithms remain very useful and indispensable.

The constraints in a GCP are very easy to be expressed and understood as saying that any two adjacent vertices cannot be assigned the same color. However, these simple explicit constraints between adjacent vertices imply many complex constraints among non-adjacent vertices. It appears that most state-of-the-art GCP algorithms do not specially detect and exploit these implicit constraints. One way to discover and exploit the implicit constraints is to encode a GCP using  $k$  colors into the Boolean satisfiability problem (SAT) by expressing the explicit constraints in CNF (Conjunctive Normal Form) clauses. Then modern SAT algorithms discover and exploit the implicit constraints among non-adjacent vertices using Conflict Driven Clause Learning (CDCL). However, the SAT instance encoding the GCP generally is very large, limiting the size of the GCP that can be solved in this way.

In recent approach researchers have tried to solve the problem using adjacency matrix based approach for the graph [16]. The same problem has address by another group of researchers and have given the solution of the problem using genetic algorithm and fuzzy logic [17]

## 3. GRAPH COLOURING

Graph coloring is a popular topic in combinatorics. There are numerous variants of colorings and various related problems. We only briefly review concepts used in this synopsis.

Let  $G$  be a graph and  $C$  be a set of colors. A coloring is a mapping  $\phi : V(G) \rightarrow C$  such that for every  $u, v \in E(G)$  holds  $\phi(u) \neq \phi(v)$ . Sometimes, it is called a proper coloring. Usually, we are interested only in proper colorings and minimizing the number of colors used. If a graph  $G$  can be colored using  $k$  colors we say that  $G$  is  $k$ -colorable. The smallest  $k$  for which  $G$  is  $k$ -colorable is called the chromatic number and it is denoted by  $\chi(G)$ . A graph  $G$  is  $k$ -critical if  $\chi(G) = k$  but every proper

subgraph of  $G$  is  $(k - 1)$ -colorable.

For example suppose a given graph as shown in figure 1 is a two colourable graph.

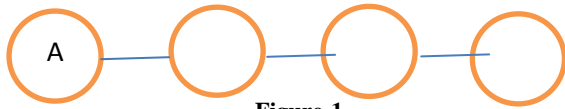


Figure-1

Most of the time researchers have focused to solve the problem by applying certain heuristics at the time of assigning the color to a particular vertex in the graph and then the color from the color list of adjacent vertices. In this process if any conflict has arisen they have tried to remove the conflict either using back tracking or some other technique. In some of the cases where the graph is dense this scheme of assigning the color to vertices works good but if the graph is a sparse graph then most of the these algorithms does not work satisfactorily.

In the figure-1 suppose there are two colors available red & black and if we want to assign the color to vertices using its applicable constraints only, the following procedure may be followed:

- 1- If vertex A has assign color red then vertex C or D can assign color red also.
- 2- Now suppose if vertex D has assigned the color red then C can be assigned only black color.
- 3- In this vertex B cannot assign any color.

Keeping the above color assignment scheme we have reached to state where conflict has arisen. To avoid such a conflict in literature another approach has find where researchers have given solution by using the concept of conflict driven clause learning management CDCL[18]. But in this this approach the following observations were found.

- 1- Although CDCLGCP effectively reduce the search space as compared to backGCP but overhead required for this in clause learning & management of learnt clause cannot be compensated against it for the random graph.
- 2- When the algorithm run on DIMACS graph then the clause learning reduce the serach tree size in CDCLGCP as compared to backGCP and it is more significant & effective to backGCP( it solve 9 more instance with in the cutoff time)
- 3- Experimental result shown that clause learning technique is not suitable for random graph due to SAT solver based on clause learning is highly inefficient for random SAT instances.
- 4- This experiment also shown that approximation algorithms are more efficient to find the upper bound on  $X(G)$  for random & quasi random graphs but not efficient in finding the LB for above graphs. However CDCL GCP algorithm can also find the LB for the above graphs.
- 5- The above proposed algorithm is efficient in the case of structured or the dense graph because these graph may contains implicit constraints and for colouring of the vertices will required to keep track the assigned colour on non-adjacent nodes. However if the graph is sparse graph in which most of vertices are independent to each other , then proposed algorithm is not so efficient as it generates conflict driven clauses and by encoding it into SAT

solver which does not required most of the time in sparse graph because there will be least chances of generation of implicit constraints but this algorithm explore non adjacent vertices for and generating the learnt clauses & process these generated clauses requires extra efforts & time which does not need in sparse graph . This may be probably one of the reasons that this algorithm is not so efficient in case of random graph.

#### 4. PROPOSED ALGORITHM FOR THE PROBLEM

- 1- The first explored node in graph  $G(V, E)$  can assign any colour.
- 2- A- The successor of assigned colour node will assigned different colour than its predecessor (by taking into consideration of explicit constraints).
- B- if there are more than one successor node and all are independent to each other ,then the same colour can assign to every node **otherwise** all different colours will assigned to successor nodes.
- C- if any node in graph is disconnected then assigned any used colour to that node.
- 3- At every node try to assign least recently used colour.
- 4- Repeat step 2 & 3 until all the vertices are assigned some colour.
- 5- Exit.

#### 5. WORKING PROPOSED ALGORITHM

The proposed algorithm shall work in the following manner:

1. Suppose the starting vertex in the given graph in figure 1 for coloring is A and there are two available colors (Red, Green) which will used to color the vertices in the graph.
2. Assign vertex A as color Red. All the adjacent vertices of A will be assigned different than A's color (step 2) and in this case there is the vertex is B and corresponding assigned color should be Green.

In the similar manner the final assigned color to vertices of the figure1 may be represented in the following table1.

Table-1

Vertex	Assigned Color
A	Red
B	Green
C	Red
D	Green

#### 6. ANALYSIS OF THE PROPOSED ALGORITHM

The above propose algorithm can be implemented with any kind of data structure suitable for graph representation that is either using linked representation of graph or by using adjacency matrix. If there are  $n$  number of vertices in the given graph and suppose there will maximum  $m$  number of adjacent vertex of a particular node in the graph then the in

step 2 it takes  $O(mn)$  time to execute.

So if the given graph is  $K$ -colorable graph then it will take maximum  $O(Kmn)$  execution time for coloring the vertices of entire graph apart from the time that will be taken for updating the color list corresponding to every vertex of the graph.

## 7. CONCLUSION AND FUTURE SCOPE

In the paper we have tried to address the graph coloring problem and proposed a solution which will work efficiently in the case of any kind of graph either it is dense graph or sparse graph. The proposed algorithm may be implemented by choosing any graph representation techniques without affecting its complexity. Since graph coloring problem is a class of optimization problem, so by the use of certain heuristics like DSATUR or some other technique at the time of exploring the vertices for coloring may improve its running time.

## 8. REFERENCES

- [1] Gamache M, Hertz A, Ouellet JO. "A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding". In *Comput Oper Res* 2007;34(8): 2384–95.
- [2] Zufferey N, Amstutz P, Giaccari P. "Graph coloring approaches for a satellite range scheduling problem". In *J Schedul* 2008;11(4):263–77.
- [3] Werra DD. "An introduction to time tabling". In *Eur J Oper Res* 1985;19:151–62.
- [4] Burke EK, McCollum B, Meisels A, Petrovic S, Qu R. "A graph based hyper heuristic for time tabling problems". In *Eur J Oper Res* 2007;176:177–92.
- [5] Werra D D, Eisenbeis C, Lelait S, Marmol B. "On a graph theoretical model for cyclic register allocation". In *Discret Appl Math* 1999;93(2–3):191–203.
- [6] Smith DH, Hurley S, Thiel SU. "Improving heuristics for the frequency assignment problem". In *Eur J Oper Res* 1998;107(1):76–86.
- [7] Woo T K, Su SYW, Wolfe RN. "Resource allocation in a dynamically partitionable bus network using a graph coloring algorithm". In *IEEE Trans Commun* 2002;39:1794–801.
- [8] Méndez-Díaz I, Zabala P. "A cutting plane algorithm for graph coloring". In *Discret Appl Math* 2008;156(2):159–79.
- [9] Lucet C, Mendes F, Moukrim A. "An exact method for graph coloring". In *Comput Oper Res* 2006;33(8):2189–207.
- [10] Méndez-Díaz I, Zabala P. "A branch-and cut algorithm for graph coloring". In *Discret Appl Math* 2006;154:826–47.
- [11] Malaguti E, Monaci M, Toth P. "An exact approach for the Vertex Coloring Problem". In *Discrete Optim* 2011;8(2):174–90.
- [12] Segundo PS. "A new DSATUR based algorithm for exact vertex coloring". In *Comput Oper Res* 2012;39:1724–33.
- [13] Chaitin GJ. "Register allocation and spilling via graph coloring". In *SIGPLAN Not* 2004;39(4):66–74.
- [14] Caramia M, Dell'Olmo P. "Coloring graphs by iterated local search traversing feasible and infeasible solutions". In *Discret Appl Math* 2008;156(2):201–17.
- [15] Josephine et.al "The Application of Graph Theory to Sudoku" *Hang Lung Mathematics Awards Vol 6(2014)*, pp 321-349
- [16] Charu Negi et.al "Vertex coloring problem Approach using Adjacency matrix" *IJETSR vol 4 issue 12,(2017)*
- [17] Tabiya Manzoor Beigh, Girdhar Gopal . Use of Genetic Algorithm and Fuzzy Logic in Optimizing Graph Coloring Problem in *IJCA Proceedings on Recent Innovations in Computer Science and Information Technology RICSIT 2016(1):1-4, September 2016*.
- [18] Zhaoyang Zhou, Chu-Min Li, Chong Huang, Ruchu Xu An exact algorithm with learning for the graph coloring problem *Computers & Operation Research* 51(2014)282-301