EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

A Greedy Technique Based Improved Approach to Solve Graph Colouring Problem

Ajay Narayan Shukla*, Vishal Bharti, M. L. Garg

DIT University, Dehradun-India, Department of CSE

Abstract

Graph colouring problem is a well-known NP-class optimization problem, studied due to a lot of applications in various real-world problems. Some of these applications are: register allocation, image processing and communication networks. There are various techniques suggested by the researchers to solve the problem which is either exact or approximate in nature. In this paper, a new greedy technique, based on degrees of vertices in the graph is presented to solve the graph colouring problem in an improved manner. The technique involves the use of adjacency matrix along with another matrix generated for the set of possible colours for each vertex in the graph. The generated colour matrix is used to assign the colours among the vertices in the graph based on decreasing degrees of the vertices. Several DIMACS colouring instances solved using the suggested approach in the article and compared with some contemporary techniques for the performance and proves compatible and having better execution time with compared technique. The obtained colouing results are mostly optimal colour values corresponding to the examined colouring instances of the graph.

Keywords: Graph Colouring, Adjacency Matrix, Greedy Technique, Colour Matrix.

Received on 21 October 2020, accepted on 28 January 2021, published on 16 February 2021

doi: 10.4108/eai.16-2-2021.168716

1. INTRODUCTION

In recent years graph colouring problem is one of the most recurrent problems studied in the literature due to its capability to address a lot of application problems in the area of Engineering, communication networks and in other real-world problems. The graph colouring problem involves the mapping of the available distinct colours among the nodes in the graph so that neighbouring nodes must be allocated with distinct colours. The minimum number of colours involved in the colouring process in the graph is called the chromatic number of the graph.

An application of graph colouring problem is found in the crew scheduling. Researchers proposed encoding of the monthly crew scheduling problem into a graph colouring problem to find a better solution of it [1]. In the article, the researchers have suggested using tabu technique of graph colouring problem that can be employed in the crew scheduling problem as well. The tabu technique can be used along with the heuristics in a sequential manner. In the proposed solution the authors embedded the branch and bound method along with a column generation technique to achieve the improved result for the problem.

The graph colouring problem can also be mapped to the register allocation in the computer. In programming, registers are allocated to different variables found in loops during the time for which proper and optimize allocation is required. This problem is also addressed using graph colouring and researchers find a solution for allocation the allocation of registers [2]. Initially, the allocation problem initiated with colouring of the vertices of a circular arc graph. The register allocation problem can be understood in terms of graph colouring by considering the variables as the vertices of the graph. The connection between the vertices in a graph implies that the variables in the particular program are required to use the same registers. The researchers suggested an improved solution for the allocation of the registers by encoding the problem into colouring problem of graph.

Another application of the graph colouring problem is found in the time table scheduling. For example time table scheduling in a college or a university where we need to schedule courses among different sections. Designing a

*Corresponding author. Email: ajayshuk@gmail.com time table with the limited resources involves a lot of complications and therefore this problem handled manually. By automating this process, would not only save time but can also eliminate the possibility of errors. Several techniques to resolve this problem have proposed and out of those one technique involves the use of genetic algorithms by transforming the problem into graph colouring. With the use of the genetic algorithm, researchers proposed an efficient solution to this problem by encoding it into graph colouring for successful mapping of the courses into the time table [3].

Another use of graph colouring problem comes from its use in address sorting in mailing process. In the business mail delivery process, the mails need to be delivered at the addresses mentioned on the envelopes. In the process, the mails which received at the post office are first sorted based on the address mentioned on their respective addresses using the OCR system. Generally, the OCR system only makes use of linear architecture which just localizes the address block. This process is not very efficient and thus has found its way to be optimized by using various techniques. One of those includes the use of graph colouring based solution. The investigators suggested a model based on hierarchical graph colouring problem to resolve the issue related to block localization and generate an improved solution for it [4].

Just like time table scheduling, examination scheduling problem can also be addressed using graph colouring technique. The constraints related to the feasible courses for which students registered along with feasible dates on which examination may be scheduled for a particular course so that common students may appear in the examination. The entire scheduling of the examination should be done efficiently with the least possible duration for scheduling the examination of all the courses. Investigators suggested the solution of the problem by encoding constraints of examination in graph colouring problem and gave the solution by creating the clusters of the nodes and then applied heuristic to get the solution of the problem [5].

There are many more applications of the graph colouring problem in other areas like personal safety issues in nuclear applications[6], frequency assignment problem in communication networks[7], resource allocation in bus networks[8], sharing of channels in dynamic spectrum network [9], caching of files near to small base station [10], minimization of spatial tube collision in video browsing [11], resource allocation in LTE networks [12], sparse matrix compression [13], spectrum utilization in femtocell network [14], role assignment in sensor network [15], in robot design [16] and in electronic equipment partitioning [17] are several key application areas addressed using graph colouring problem.

In the proposed research work, we present a greedy technique for colouring of the vertices based on the computation of the degree of each vertex in the graph and then using computed information and applying the colouring procedure to allocate the colour to each vertex in the graph. The technique involves several other auxiliary procedures to perform the colouring process which is exact in nature with improved execution time.

Although the graph colouring problem can be addressed using several optimization techniques but we tried to solve the problem using the explicit constraint of the problem with the help data structure available in the literature. Since the running time of the operations performed on the used data structure is most of the polynomial order and therefore the proposed technique produces colouring solution in more optimized running time. The mechanism used to devise the algorithms for solving the problem is independent of the types of graph and therefore algorithms may be applied to any kind of graph. The colouring scheme of vertices in the graph is based on the construction of a set of vertices in the decreasing order of degree and entire procedures implemented using array data structure to handle sets generated and matrices used to store graphs and other information.

2. RELATED WORK

The allocation of channels for frequency in the communication network is addressed by transforming the problem as colouring problem of a graph then formulating the colouring problem using set covering technique, the researchers computed an optimal number of channels required to assign the frequency in the network [18]. The graphs produced in the application involves VLSI technology is another area in which the produced graphs needs to be coloured. The technique used to colour the vertices of the generated graphs is by decomposing the graph in a binary fashion. The resulted decomposition is in the form of balanced height decomposition and the authors suggested approximation based procedure using max-cut principle to solve the colouring problem of graph [19].

The computation based on the optimization of biological operations by the use of DNA Computing technique researchers suggested an optimal way to produce the colouring solution of graphs [20]. The approach is based on the proposition that vertices in the graph is ordered in such a way that mostly nth and (n+1)th labelled vertex is neighbouring nodes to each other. The second constraint used in the technique is related to the use of the least number of encodings representing the colours in the problem. The applied technique is examined by conducting the experiment with a graph of twelve vertices and the result obtained with three colours which was the chromatic number of the examined graph. Although the experimental data set is small but it gives the new technique to address the solution for the problem that can also be applied to find the solution for graphs having a large number of nodes.

Evolutionary technique is also applied for finding the graph colouring problem's solution. In one approach that uses optimization technique involving Multiple-Resart Quasi-Hopfield network [21]. The authors identified three variants of colouring problems for finding the solution. The addressed colouring problems are as under.

i- Minimum Colouring problem
ii- Spanning Subgraph k-colouring problem
iii- Induced Sugraph k-colouring problem

For getting the efficient running time for their proposed algorithms the technique involves the use of k-state neurons instead of binary neurons that reduces the connections in the constructed neural network.

Several other approaches are available in the literature to address the problem using optimization techniques. Artificial Bee Colony optimization-based approach [22] , DNA algorithm based multi separation model [23], learning automata-based solution [24], using chaos neural network [25] in which researchers non-monotonous activation function in the proposed model to get the solution of the problem.

The researchers suggested a technique to assign the colours among the vertices in the graph using a method that is exact in nature, comprises of linear decomposition applied on the considered graph for colouring [26]. For handling the graphs with unbounded linear width they suggested a hybrid technique based on decomposition and heuristic method. The brief outlines of the procedure adopted to find the colouring solution for bounded linear width graphs is as under.

```
1. Consider the graph G(V,E) as input
2. Compute lower bound and upper bound of C
   // here C is chromatic number of the considered graph for colouring
3. Repeat
4.    Establish the value of C between the lower bound and upper bound computed in step 2
5.    Solution = C-colouring( G)
6.    If( solution = FALSE) then lower bound= lower bound+1
      Else upper bound= C
7. Until ( lower bound != upper bound)
8. End
```

In the above steps, C-colouring is an auxiliary procedure that computes the colouriing solution and it returns True value when the graph is C- colourable graph and this computation is based on the decomposition of the graph. The suggested procedure works efficiently for the graphs having a small linear width like in case mug category of graphs.

Authors also suggested the solution of the problem by transforming the graph into the trees to compute maximal independent sets so that each vertex in a particular set may be assigned similar colour [27]. Authors suggested a hybrid technique that uses the Learning Automata along to perform local search and Michigan technique to generate colouring solution [28]. The suggested procedure works in a distributed manner in which each chromosome locally grows using evolutionary operator by improving solution set using local search. The technique involves branch and pricing model for constructing the colouring solution is another approach that uses a good mix of metaheuristic, tabu search and column generation mechanism to get optimal colour values for the graph [29].

Researchers developed a model, based for the design of railway station with the optimized infrastructural capability to accommodate facilities by encoding the problem in graph colouring problem [30]. In this problem, authors tried to provide shortest infrastructure occupation time of the track from the current running time of all the tracks on the station. Improved utilization of cellular capacity in the vehicle to vehicle communication is another application area that has addressed efficiently using graph colouring problem [31]. The caching of files in the memories of small cell base station is addressed by the researchers by transforming the problem into graph colouring to find the optimized caching of the files in the memory of base station cells [32].

The versatile applications of this problem in various real-world problems always motivate the researchers to provide an improved way of finding the colouring solution of the graphs generated from the application problem.

3. MOTIVATION FOR OUR APPROACH

Graph colouring problem is studied due to its various applications in the real world problem which required to be solved efficiently not only in terms of optimal colour values but also for the result in better execution time. There is a lot of techniques available in the literature to address the problem using different techniques. But there is no single technique available in the literature that solves all different categories of colouring instances of the graphs generated from different real-world applications. Therefore we have tried devise a technique that may be used to solve any graph instances generated from application problems at least in terms of successful colouring results. The technique is also experimentally examined for several categories of DIMACS colouring instances for its running time and optimal colouring values of tested graphs

4. DATA STRUCTURE & TECHNIQUE USED FOR COLOURING

The proposed technique involved the use of adjacency matrix representation of graph that stores the adjacent node of each vertex in the graph. The adjacent node information of each vertex is retrieved from the matrix after assigning a particular colour to the designated vertex to forbid used colour to other neighbouring vertices. Let a graph drawn in

Figure 1 is a 3-colourable graph and its adjacency matrix is shown in Table 1.

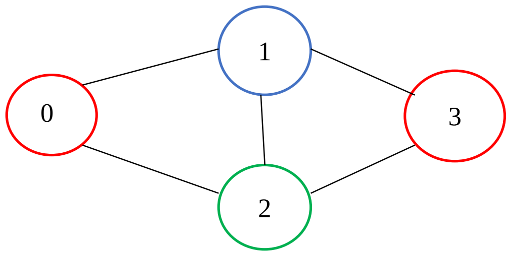

Figure1: An example of 3-colourable graph

The neighbouring information of the vertices in the above graph is stored in the matrix created in Table 1.

	0	1	2	3
0	0	1	1	0
1	1	0	1	1
2	1	1	0	1
3	0	1	1	0

Table 1: Adjacent matrix of graph of figure 1

The entries in the matrix stored in table 1 consist of binary values. The value 0 in any cell represents that the vertex in the corresponding row is non-adjacent vertex of the corresponding column (like node 0 is non-adjacent node to 3) otherwise vertex in the row is the adjacent vertex of corresponding column vertex (like node 1 is the adjacent node to vertex 3). The adjacency matrix generated for the graph is stored in a two-dimensional array of size NxN where N is the number of vertices in the graph.

We used additionally colour matrix that stores feasible colour information corresponding to each vertex in the graph. The colour matrix created for the graph in figure 1 is shown in table 2.

Vertex	**Set of feasible colours**			
0	0	1	2	3
1	0	1	2	3
2	0	1	2	3
3	0	1	2	3

Table 2: Colour matrix created for the graph of figure 1.

The objective of the above colour matrix created in table 2 is that initially, we assuming that the number of colours required to colour the nodes in the graph is maximum up to the number of vertices in the graph and it is true in case of a graph generated in the application problem is a completely connected graph. Although in this assumption the only drawback is that it consumes more memory to represent a colour matrix in case of the chromatic number of a graph is smaller than the number of nodes but its execution time is not varied so much for usual cases except for initialization phase where the entire matrix is initialized with discrete values. The marking of forbidden colours for a particular row is done only for the currently used colours that is supposed to be restricted in adjacent nodes.

5. PROPOSED ALGORITHMS

The proposed technique selects the un-coloured vertex having a maximum degree in every phase for colouring. Initially, the degree of each vertex in the graph is computed and stored in a one-dimensional array.

Algorithm Greedy_GCP(int a[][],int g[][])

```
{ //here matrix a and g are global array and stored
adjacency //matrix and feasible colour values for the
vertices respectively.
// m is number of vertices in the graph.
int l, c[], b[];
initialize_colourmatrix( );
//calculation of degrees of vertices in the graph
for(int l=0;l<m;l++)
{  b[l]=0;
for(int s=0;s<m;s++)
if(a[l][s]==1)
b[l]=b[l]+1;
}
//assignment of colours
for(int p=0;p<m;p++)
{
int max= -1;
for(int t=0;t<m;t++)
{
if (b[t]>=0 &&b[t]>=max)
{
  max=b[t];
  i1=t;
}
}
  b[i1]= -2;

if(g[i1][0]!= -2)
{
k=c[i1]=search(i1);
g[i1][0]= -2;
}
for(int q=0;q<m;q++)
{
if(a[i1][q]==1)
update(q,k);
}
for(q=0;q<m;q++)
{
if(a[q][i1]==1)
update(q,k);
}
}
```

The algorithm Greedy_GCP( ) computes the degrees of each vertex in the graph initially before the actual colouring process. After that procedure discovers the un-coloured vertex with the highest degree for colouring. Then it calls the procedure search to select the appropriate colour for the selected vertex for colouring. Finally, the procedure terminates with updating the colour matrix g for adjacent nodes of the currently coloured vertex.

Algorithm initialize_colourmatrix()

```
{ // initialize initial colour matrix
for(int i=0;i<m;i++)
g[i][0]=i;
for(i=0;i<m;i++)
for(int j=1;j<m+1;j++)
g[i][j]=j-1;
}
```

The above procedure initialize_colourmatrix ( ) called only a single time in Greedy_GCP( ) procedure to initialize the colour matrix with the maximum possible number of feasible colours to each vertex in the graph.

Algorithm search (int i)

```
{ // find appropriate colour for vertex i
for(int j=1;j<m;j++)
if(g[i][j]>=0)
return(g[i][j]);
}
```

The above algorithm search ( ), return one feasible colour from the colour matrix g[][] that is retrieved first non-negative column value from the matrix corresponding i^{th} row.

Algorithm update (int i,int k1)

```
{ // forbid colour k1 to vertex i
for(int j=1;j<m+1;j++)
{
if(g[i][j]==k1)
   g[i][j]= -1;
   }
 }
```

The algorithm update forbid the colour value k1 assigned to currently coloured vertex by replacing with -1 to adjacent vertex i of currently coloured vertex i1 in Greedy_GCP( ).

6. ANALYSIS OF THE PROPOSED ALGORITHMS

The formal analysis of the suggested procedure is straightforward and simple. The algorithm Greedy_GCP ( ) started with the one-time initialization of the colour matrix followed with the computation of degrees of the vertices in the graph. In the later phase of the algorithm each vertex selected for colouring in the decreasing order of degrees. Finally, after allocation of the colour to a particular vertex, the update procedure called for each adjacent vertex of currently coloured vertex to forbid the used colour. The run time analysis of the suggested procedures is computed in the following way.

1. The algorithm initialize_colourmatrix ( ) takes $O(m^2)$ running time where m is the number of vertices in the graph.
2. The computation of degrees for each vertex in the graph takes $O(m^2)$ running time.
3. The selection of vertex with the highest degree computed in O(m) running time and since this executed in Greedy_GCP ( ) maximum m times, therefore, the total complexity will be $O(m^2)$.
4. The search() procedure called m times in the Greedy_GCP () and in each of its call the algorithm executed maximum m times(in case of the graph is a clique). Therefore the total complexity will be $O(m^2)$.
5. Finally, the update procedure called in Greedy_GCP ( ) 2*m*m times. Each call of update () procedure involves running time of maximum O(m). So the total running time in this process will be $O(m^3)$.

On cumulating all the running times involved in the entire procedures the final running time of all the procedures will be $O(m^3)$.

7. ILLUSTRATION WITH EXAMPLE

Suppose the graph shown in figure 1 is coloured using the suggested algorithms in section 4 that requires initially a colour matrix as constructed in table 2. The number of degree corresponding to each vertex of the graph is computed in table 3

Vertex	Degree
0	2
1	3
2	3
3	2

Table 3: The degrees of the vertices in the graph of Figure 1

Now the vertex with the highest degree is selected for colouring. In the case of few vertices having the same maximum values the last vertex found is selected for colouring. Like in above table 3 vertex 1 and vertex 2 is having the same highest degree 3 and therefore the vertex 2 is selected for colouring.

The first non-negative colour available in colour matrix corresponding row containing vertex 2 is returned by the search procedure for the allocation to node 2 in the graph. The updated colour matrix after assigning the colour to node 2 is computed in table 4. Similarly, table 3 that contains the degree information for each vertex is also updated by resetting the number of the degree corresponding to currently assigned colour vertex with value -2. The negative degree of a vertex in the table indicates that the vertex is already assigned some colour in previous iterations. The updated table for the degree of vertices is shown in table 5. To forbid a particular colour from the adjacent nodes of currently assigned colour vertex the colour value is reset with value -1.

Vertex	Set of feasible colours			
0	**-1**	1	2	3
1	**-1**	1	2	3
-2	0	1	2	3
3	**-1**	1	2	3

Table 4: Updated colour matrix after allocation of colour 0 to vertex 2.

In the above table, -1 entry corresponding to each row indicates that colour 0 cannot be assigned to nodes 0,1and node 2 respectively. The reason for making vertex as -2 is that at the time of updating the colours for other vertices this node will not be explored as previously it has already allocated some colour.

Vertex	Degree
0	2
1	3
2	**-2**
3	2

Table 5: Updated degrees of the vertices in the graph after assigning the colour to vertex 2.

Similarly in the above table containing the degree of vertex 2 is reset with value -2 indicates that this node has assigned some colour and therefore in the next phase of finding the node with a maximum degree, this node is automatically discarded.

Finally, the vertex selected for colouring, set feasible colours and colour allocated to a particular vertex is shown in table 6.

Vertex selected for colouring	Feasible colours	Assigned colour
2	0,1,2,3	0
1	-1,1,2,3	1
3	-1.-1,2,3	2
0	-1,-1,2,3	2

Table 6: Sequence of vertex selected for colouring based on decreasing order of the number of degrees of the graph of figure 1.

In the above table, the set of colour values is considered from 0 to m-1 in the increasing order and m is the number of vertices in the graph. The negative colour values indicate that these colour sequences have forbidden to the corresponding vertex.

8. EXPERIMENTAL OUTCOMES AND DISCUSSIONS

The algorithms presented in section 5, have examined on the several graph colouring instances of DIMACS [37]. These graphs are in the form of (vertex, edge) pair (see at https://mat.tepper.cmu.edu/COLOR02). The algorithms executed on Turbo C3 2.2.0 compiler on Windows 7(32 bit) platform with Intel Pentium D CPU@2.8 GHZ, 4 GB RAM. The tested graphs is stored in the form of an adjacency matrix by converting the DIMACS instances using Python 3.6.

The proposed algorithms terminated with optimal colouring values for most of the tested graphs. The DIMACS graph colouring instances and the associated result on the execution of the proposed algorithms are computed in Table 7. In the table the first column indicates the name of graph, subsequently, 2nd column contains the number of vertices in the graph, 3rd column contains the number of edges in the graph, 4th column contains colouring results (K) after the execution of the proposed algorithms and 5th column contains execution time in seconds. X (G) is the chromatic number of the graph and entry in the 6th column as "?" indicates that chromatic number is unknown and 7th column shows the best-known colouring values for the graphs of the corresponding 1st column.

Graph Name	|V|	|E|	K	Time (S)	X (G)	Best known K
myciel3	11	20	4	0.000	4	4
myciel4	23	71	5	0.000	5	5
myciel5	47	236	6	0.165	6	6
myciel6	95	755	7	0.385	7	7
myciel7	191	2360	8	1.318	8	8
david	87	812	11	0.440	11	11
huck	74	602	11	0.275	11	11
jean	80	508	10	0.275	10	10
anna	138	986	11	0.824	11	11
mugg88_1	88	146	4	0.330	4	4
mugg88_25	88	146	4	0.330	4	4
2-Insertions3	37	72	4	0.055	4	4
3-Insertions3	56	110	4	0.220	4	4
1-Insertions4	67	232	5	0.275	4	5
1-Insertions5	202	1227	6	1.813	?	6
2-FullIns3	52	201	5	0.220	?	5
4-FullIns3	114	541	7	0.549	?	7
3-FullIns3	80	346	6	0.549	?	6
5-FullIns3	154	792	8	1.209	?	8
miles250	128	774	8	0.879	8	8
miles500	128	2340	20	1.154	20	20

miles1500	128	10396	73	2.527	73	73
zeroin.i.1	211	4100	49	2.747	49	49
zeroin.i.2	211	3541	30	2.582	30	30
zeroin.i.3	206	3540	30	2.527	30	30
mulsol.i.1	197	3925	49	2.418	49	49
mulsol.i.2	188	3885	31	2.308	31	31
mulsol.i.3	184	3916	31	2.253	31	31
mulsol.i.4	185	3946	31	2.368	31	31
mulsol.i.5	186	3973	31	2.308	31	31

Table 7: Colouring results on the execution of proposed algorithms.

The above colouring result is compared with several other existing techniques and in most of the cases our proposed technique is found comparable and in some cases, our result is better either in execution time or in terms of optimal colouring values.

On comparing with the modified cuckoo optimization technique [33] for solved DIMACS graph colouring instances the result is shown in table 8.

Graph Name	**K(our case)**	**Time(our case)**	**K[33]**	**Time [33]**
myciel3	4	**0.000**	4	0.2
myciel4	5	**0.000**	5	0.3
myciel5	6	**0.165**	6	0.3
myciel6	7	**0.385**	7	0.5
myciel7	8	**1.318**	8	3.8
david	11	**0.440**	11	0.5
huck	11	**0.275**	11	0.5
jean	10	**0.275**	10	0.5
anna	11	0.824	11	0.8
mugg88_1	4	**0.330**	4	1.1
mugg88_25	4	**0.330**	4	1.3
2-Insertions3	4	**0.055**	4	0.4
3-Insertions3	4	**0.220**	4	0.5
1-Insertions4	5	**0.275**	5	0.5
1-Insertions5	6	1.813	6	1.2
2-FullIns3	5	**0.220**	5	0.4
4-FullIns3	7	**0.549**	7	0.7
3-FullIns3	6	0.549	6	0.4
5-FullIns3	8	1.209	8	0.5
miles250	8	**0.879**	8	1.1
miles500	20	**1.154**	20	1.2
miles1500	73	2.527	73	1.2
zeroin.i.1	49	2.747	49	1.7
zeroin.i.2	30	2.582	30	2.1
zeroin.i.3	30	2.527	30	1.6
mulsol.i.1	49	2.418	49	1.2
mulsol.i.2	31	2.308	31	1.1
mulsol.i.3	31	2.253	31	1.3
mulsol.i.4	31	2.368	31	1.7
mulsol.i.5	31	2.308	31	1.9

Table 8: Comparison results with our case & MOCA [33]

On analyzing the comparison results shown in table 8, the colouring results for compared graphs are the same but in a few cases, the running time of our proposed technique indicated in bold is better than MOCA. Although the hardware configuration used for experimentation in MOCA is unknown and therefore the execution mentioned in the research article is used for comparison.

The comparison result with a technique based on edge cover-based graph colouring [34] and phase dynamics coupled oscillatory networks [35] finds more promising in terms of optimal colouring values for the tested graphs. The comparison results are shown in Table 9.

Graph Name	**K(our case)**	**K[34]**	**K[35]**
myciel3	4	4	4
myciel4	5	5	5
myciel5	6	6	6
myciel6	**7**	7	**8**
myciel7	8	**10**	-
david	11	**12**	**13**
huck	11	11	**12**
jean	10	10	-
anna	11	**12**	-
mugg88_1	4	**5**	-
mugg88_25	4	4	-
2-Insertions3	4	4	-
3-Insertions3	4	4	-
1-Insertions4	5	5	-
1-Insertions5	6	6	-
2-FullIns3	5	5	-
4-FullIns3	7	**8**	-
3-FullIns3	6	6	-
5-FullIns3	8	8	-
miles250	8	**10**	-
miles500	20	**26**	-
miles1500	73	**81**	-
zeroin.i.1	49	**51**	-
zeroin.i.2	30	**32**	-
zeroin.i.3	30	**32**	-
mulsol.i.1	49	49	-
mulsol.i.2	31	31	-
mulsol.i.3	31	31	-
mulsol.i.4	31	31	-
mulsol.i.5	31	31	-

Table 9: Comparison results with [34] and [35]

The comparison results are shown in Table 9 with edge cover-based graph colouring technique our greedy technique based approach is found better and produces optimal colouring values or near to optimal colouring for 11 graphs mentioned in bold entries in 3rd column [34] in the table. Similarly on comparing with another technique reported in the literature in which researchers used oscillatory circuits for encoding the colouring problem for getting the solution our technique finds more promising and achieved the optimal colouring values for the compared graphs and other fails to achieve that. For the graphs myciel6, david and huck the colouring values in the 4th of the table is optimal in our case however for [35] it is not so. Blank entries in the 4th column indicate that the graphs corresponding to those columns not considered in the paper.

On comparing the result obtained in greedy based improved technique for graph colouring problem with modified simulated annealing technique [36] our result is better in terms of execution time for each compared graph instances (Table 10). In a few class of graphs, our technique produces optimal or near to optimal colouring values (bold entries in 2nd column) but modified simulated annealing technique fails to achieve optimal colour values for those graphs

Graph Name	K(our case)	Time(S) in our case	K MSAGP [36]	Time (s) [36]
myciel6	7	0.385	7	6683
david	11	0.440	11	2972
huck	11	0.275	11	31
jean	10	0.275	10	45
anna	11	0.824	11	2535
mugg88_25	4	0.330	4	60
1-Insertions5	**6**	1.813	7	786
4-FullIns3	7	0.549	7	127
5-FullIns3	8	1.209	8	517
miles1500	**73**	2.527	75	5885
zeroin.i.2	30	2.582	30	2341
mulsol.i.1	**49**	2.418	52	2861

Table 10: Comparison results with modified simulated annealing [36]

Although the hardware used in [36] for the implementation of the suggested algorithm by the researchers Xeon@2.4 GHZ with 1 GB RAM which is a little bit slower than our case (CPU@ 2.8 and 4 GB RAM) but there is a significant difference between running time with our case and [36] and our technique is more efficient as compared to modified simulated annealing technique for solving the graph colouring problem.

9. CONCLUSION AND FUTURE SCOPE

The proposed work intends to solve the graph colouring problem based on the greedy selection of vertex in the decreasing order of degree for colouring. The algorithms proposed can be applied to any kind of graphs for finding the colouring solution in an exact manner. The computed complexity for the procedures results in polynomial order and on most of the tested DIMACS graph colouring instances our technique produces optimal colouring values.

Although there is no single technique available in the literature that can solve all categories of the graphs with optimal colour values. Therefore in the presented technique we used data structure available in the literature and tried to solve the problem with efficient running time for those problems that may be successfully solved with our technique.

Since this graph colouring problem is an NP-complete problem and therefore there is always a hope that the running time of the contemporary techniques may be optimized. The other improvement may be possible in terms of memory uses in the proposed technique. Like in our case initially colour matrix uses m number of colours which is only possible in case the graph is completely connected graph. Another improvement may be possible by eliminating the requirement of colour matrix that reduces the memory requirement as well as its processing time for keeping the previous assigned colour information of the vertices.

References

[1] Gamache M, Hertz A,Ouellet JO. ,"A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding", Computers & Operations Research, Vol. 34, No. 8, pp. 2384-2395, 2007.

[2] Werra D D, Eisenbeis C, Lelait S, Marmol B. ,"On a graph theoretical model for cyclic register allocation", Discrete Applied Mathematics, Vol. 93, No. 2-3,pp. 191-203,1999.

[3] Maram Assi, Bahia Halawi, Ramzi A. , Haraty, " Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem",Procedia Computer Science ,Vol. 216, pp. 899-906, 2018.

[4] Djamel Gaceb, Véronique Eglin, Frank Lebourgeois,, Hubert Emptoz, "Robust Approach of Address Block Localization in Business Mail by Graph Coloring", The International Arab Journal of Information Technology, Vol. 6, No. 3,pp. 221-230, 2009.

[5] B. Hussin, A. S. H. Basari, A. S. Shibghatullah, S. A. Asmai and N. S. Othman, "Exam timetabling using graph colouring approach", 2011 IEEE Conference on Open Systems, Langkawi, pp. 133-138, 2011.

[6] Mohammed Khasawneh, Mohamad I. Malkawi ,"A Graph-Coloring-Based Navigational Algorithm for Personnel Safety in Nuclear Applications", Proceeding of the 6thInternational Symposium on Mechatronics and its Applications (ISMA09), Sharjah, UAE, pp. 23-26, 2009.

[7] Smith DH, HurleyS, Thiel SU.," Improving heuristics for the frequency assignment problem", European Journal of Operational Research, Vol. 107,No. 1, pp.76-86 ,1998.

[8]] WooTK, Su SYW, Wolfe RN. ,"Resource allocation in a dynamically partitionable bus network using a graph coloring algorithm", IEEE Transactions on Communications, Vol. 39, No.1, pp. 794-801, 2002.

[9] N. Kumar H.S. and M. Chatterjee, "Scheduling in dynamic spectrum access networks using graph coloring," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, 2015, pp. 20-23.

[10] M. Javedankherad, Z. Zeinalpour-Yazdi and F. Ashtiani, "Cache Placement Phase Based on Graph Coloring," 2018 9th International Symposium on Telecommunications (IST), Tehran, Iran, 2018, pp. 187-191.

[11] T. Ruan, S. Wei, J. Li and Y. Zhao, "Rearranging Online Tubes for Streaming Video Synopsis: A Dynamic Graph Coloring Approach," in IEEE Transactions on Image Processing, vol. 28, no. 8, pp. 3873-3884, 2019.

[12] D. Tsolkas, E. Liotou, N. Passas and L. Merakos, "A graph-coloring secondary resource allocation for D2D communications in LTE networks," 2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Barcelona, 2012, pp. 56-60.

[13] H. M. Bücker, M. A. Rostami and M. Lülfesmann, "An interactive educational module illustrating sparse matrix compression via graph coloring," 2013 International Conference on Interactive Collaborative Learning (ICL), Kazan, 2013, pp. 330-335

[14] Qian Zhang, Xinning Zhu, Leijia Wu and K. Sandrasegaran, "A coloring-based resource allocation for OFDMA femtocell networks," 2013 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, 2013, pp. 673-678

[15] L. M. Rizzo, S. Urrutia and A. A. F. Loureiro, "Role Assignment in Wireless Sensor Networks Based on Vertex Coloring," 2013 IEEE Seventh International Symposium on Service-Oriented System Engineering, Redwood City, 2013, pp. 537-545

[16] S. Mikami, T. Fukuda and S. Suzuki, "Gait design by graph coloring for robots that have legs on their faces," 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Busan, 2015, pp. 116-120.

[17] V. V. Kureichik, V. V. Kureichik and D. V. Zaruba, "Partitioning of ECE schemes components based on modified graph coloring algorithm," Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2014), Kiev, 2014, pp. 1-4.

[18] S. H. Cameron, "The Solution of the Graph-Coloring Problem as a Set-Covering Problem," in IEEE Transactions on Electromagnetic Compatibility, Vol. EMC-19, No. 3, pp. 320-322, 1977.

[19] Jun-Dong Cho, S. Raje and M. Sarrafzadeh, "Fast approximation algorithms on maxcut, k-coloring, and k-color ordering for VLSI applications", IEEE Transactions on Computers, Vol. 47, No. 11, pp. 1253-1266,1998.

[20] J. Xu et al., "An Unenumerative DNA Computing Model for Vertex Coloring Problem", IEEE Transactions on NanoBioscience, Vol. 10, No. 2, pp. 94-98, 2011.

[21] A. Di Blas, A. Jagota and R. Hughey, "Energy function-based approaches to graph coloring" , IEEE Transactions on Neural Networks, Vol. 13, no. 1, pp. 81-91, 2002.

[22] R. S. Tomar, S. Singh, S. Verma and G. S. Tomar, "A Novel ABC Optimization Algorithm for Graph Coloring Problem", 5th International Conference and Computational Intelligence and Communication Networks, Mathura, pp. 257-261, 2013.

[23] L. Wang, "A DNA Algorithm of Graph Vertex Coloring Problem Based on Multi-Separation," 2009 WRI Global Congress on Intelligent Systems, Xiamen, 2009, pp. 547-550.

[24 J. A. Torkestani and M. R. Meybodi, "Graph Coloring Problem Based on Learning Automata," 2009 International Conference on Information Management and Engineering, Kuala Lumpur, 2009, pp. 718-722.

[25] X. Wang and Q. Qiao, "Solving Graph Coloring Problems Based on a Chaos Neural Network with Non-monotonous Activation Function," 2009 Fifth International Conference on Natural Computation, Tianjin, 2009, pp. 414-417.

[26] Lucet C, Mendes F, Moukrim A. ,"An exact method for graph coloring", Computer & Operations Research, Vol. 33,No. 8,pp. 2189–2207, 2006.

[27] Patidar H., Chakrabarti P(2019) ,"A Tree-Based Graph Coloring Algorithm Using Independent Set," Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 714. Springer, Singapor.

[28] Mehdi Rezapoor Mirsaleh, Mohammad Reza Meybodi," A Michigan memetic algorithm for solving the vertex coloring problem," Journal of Computational Science, vol. 24, pp. 389-401, Jan. 2018.

[29] Malaguti E, Monaci M, Toth P. ,"An exact approach for the Vertex Coloring Problem", Discrete Optimization, Vol. 8, No. 2, pp. 174-190, 2011.

[30] Predrag Jovanović, Norbert Pavlović, Ivan Belošević, Sanjin Milinković,"Graph coloring-based approach for railway station design analysis and capacity determination", European Journal of Operational Research, Vol. 287, No. 1, pp.348-360-2020.

[31] S. Xiao, W. Li, L. Yang and Z. Wen, "Graph-Coloring Based Spectrum Sharing for V2V communication," *2020 International Conference on UK-China Emerging Technologies (UCET)*, Glasgow, United Kingdom, 2020, pp. 1-4, doi: 10.1109/UCET51115.2020.9205455.

[32] M. Javedankherad, Z. Zeinalpour-Yazdi and F. Ashtiani, "Content Placement in Cache Networks Using Graph Coloring," in *IEEE Systems Journal*, vol. 14, no. 3, pp. 3129-3138, Sept. 2020, doi: 10.1109/JSYST.2020.2978105.

[33] Shadi Mahmoudi, Shahriar Lotfi, "Modified cuckoo optimization algorithm (MCOA) to solve graph coloring problem", Applied Soft Computing, Vo. 33, pp. 48 – 64, 2015.

[34] Patihar H,Chakrabarti ," A Novel Edge Cover based Graph Coloring Algorithm", IJACSA, Vol. 8,No.5, p.p. 279-286, 2017.

[35] Abhinav Parihar et al. "Vertex coloring of graphs via phase dynamics of coupled oscillatory networks". Scientific Reports, 7:11, DOI: 10.1038/s41598-017-00825-1

[36] Pal Jotyi Aninda et al "Comparative Performance of Modified Simulated Annealing with Simple Simulated Annealing for Graph Coloring Problem", Procedia Computer Science 9(2012) pp 321-327.

[37] https://mat.tepper.cmu.edu/COLOR02/