Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 7s (2025)

Efficient Edge Chromatic Number Calculation Using Python: A Matrix Based Approach

C. Paul Shyni[1] T. Ramachandran[2] and R. Divya[3]
[1]Research Scholar, Mother Teresa Women's University, Kodaikanal, Dindigul, Tamil Nadu
Assistant Professor, Department of Mathematics, St. Antony's College of Arts and Sciences for Women, Thamaraipadi, Dindigul, Tamil Nadu.
g-mail: shinyswe@gmail.com
[2]Department of Mathematics, M.V Muthiah Government Arts College for Women, Dindigul, Tamil Nadu
g-mail: yasrams@gmail.com
[3]Department of IT, PSNA College of Engineering & Technology, Dindigul.
g-mail: divivicky2015@gmail.com

Article History:

Received: *27-10-2024*

Revised:*11-11-2024*

Accepted:*19-12-2024*

Abstract:
Graph coloring is one of the most important concept in graph theory and is used in many real time applications. Graph coloring includes, vertex coloring, edge coloring, region coloring etc., of graphs. Proper edge coloring means, coloring of edges of a graph such that no two adjacent edges have the same color. Edge chromatic number of a graph is the minimum number of colors required for proper edge coloring of the graph and is denoted by $\chi'(G)$. In this article, GEC algorithm is proposed for finding the edge chromatic number of a graph with computer coding for the same using python. This proposed computer algorithm is explained with an example and it helps to understand the execution process of python to find the edge chromatic number of a graph.

Keywords: Edge coloring, Edge Chromatic number, Edge Adjacency matrix, Edge degree, Algorithm, Python.

1. Introduction

The origin of graph theory started with the problem of Konigsberg bridge in 1735. Graph theory ideas are highly utilized by computer science applications. Graph theory has so many parameters in which graph coloring is one among them. Graph coloring is an essential concept and especially used in the research areas of computer science such as data mining, image segmentation, clustering, image capturing, networking etc., [1]. With the help of graph coloring concepts, we can solve various timetabling problems, scheduling problems, job allocation problems, team building problems, etc. There are several methods of graph coloring available that can be applied in the immense areas of science and technologies as per the condition of the problems. In most of the coloring problems, it is expected to discover the chromatic number of graphs i.e., the least possible number of colors for appropriate coloring of graph G [16]. So far there are several algorithms were proposed by various authors for vertex coloring, egde coloring, total coloring and region coloring etc. By analysing the methodologies used in the existing algorithms and knowing the importance of graph coloring, in this paper GEC algorithm is proposed to find the edge chromatic number of a graph.

2. Graph coloring algorithms- A Review

This section briefs various existing graph coloring algorithms. Greedy algorithm uses the techniques of deciding the color of vertices sequentially in the coloring process, which gives the minimum number of colors for vertex coloring but it need not be a chromatic number [2]. Tabu search techniques provide the optimal coloring of a graph having up to 1000 nodes and their efficiency is shown to be significantly superior to the famous simulated annealing [3]. David S. Johnson et al presented the simulated annealing and its parameterized generic implementation, describes how this generic algorithm was adapted to the graph partitioning problem, and reports how well it compared to standard algorithms [4]. Daniel Brelaz described efficient new heuristic methods to color the vertices of a graph which rely upon the comparison of the degrees and structure of a graph [5]. Amit Mittal et al described a method for graph coloring with minimum number of colors and it takes less time as compared to other techniques [6]. K A Santosa et al, presented in the form of the pseudocode and flowchart so that it can be computerized more easily. The novelty of this research is to detect the character of the adjacency matrix so that it can apply to vertex colouring through the matrix [7]. T. Ramachandran presented an algorithm based on the adjacency matrix, to color all the vertices of the given graph with the minimum number of colors which helps us to determine the chromatic number of any graph [8]. Samsul Arifin presented a Bitwise coloring algorithm using Python [9]. Ganga K. M. et al, presented First fit algorithm, Welsh Powell algorithm, Largest degree ordering algorithm, Incidence Degree Ordering Algorithm, Degree of Saturation Algorithm and Recursive Largest First Algorithm using Python Programming [10].

3. About Python Programming

Python is a multipurpose programming language and has lot of built-in library support. It has variety of modules and software extensions to customize the application to meet specific requirements. Python’s community is very active in developing the language so that it becomes a very trustworthy language. It has the ability to communicate with other languages and the code can be called from C, C++ and other programming languages and vice versa. Python is not only easier to read, but it is also more efficient than the languages like C, C++ and Java. Minimum number of lines can be used in writing python program when compared with other languages. Python is a widely used programming language that is considerably easier to learn. Because the idea of python itself promotes readability, the syntax is straightforward, easy to comprehend and remember. Python code is simple to develop and read, making it simple to debug and maintain. In short, the reason we use Python is that it is a powerful language and can be run on multiple platforms, also very easy to understand [9]. In Particular, Python is commonly used to implement graph coloring algorithms like the Greedy Algorithm, Backtracking, and Welsh-Powell due to its flexibility and vast library support. Python's NetworkX library provides a robust framework for handling graph structures and has built-in methods for graph coloring, making it easy to model and solve problems related to coloring [17].

4. Preliminaries

For clear and better understanding of the GEC algorithm some basic definitions and their remarks are presented below.

4.1 Definition

Coloring all the edges of a graph with colors such that no two adjacent edges have the same color is called the *proper edge coloring*.

A graph in which every edge has been assigned a color according to a proper edge coloring is called a *proper edge colored graph*.

A graph G requires k different colors for its proper edge coloring, and no less, is called a *k- chromatic graph*, and the number k is called the *chromatic number* of G.

Remarks

- A graph consisting of only isolated edges is 1-chromatic.
- A graph with one or more edges (not a self – loop) is at least 2 – chromatic.
- A complete graph of n edges is n-chromatic.
- A graph consisting of simply one circuit with n≥3 vertices is 2 – chromatic if n is even and 3-chromatic if n is odd.

4.2 Definition

Edge Adjacency Matrix of G be a graph with n vertices, e edges, and no self – loops is defined by an e by e matrix denoted by A= $[a_{ij}]$, whose e rows and e columns are corresponding to the e edges. The matrix elements are

$[a_{ij}]$ = 1, if i^{th} edge is adjacent to j^{th} edge,
= 0, otherwise

Remarks

- Edge Adjacency matrix is also known as [0,1] matrix
- Edge Adjacency matrix is symmetric,i.e $[a_{ij}] = [a_{ji}]$
- Simple graph doesn't have a loop, so the diagonal elements of the edge adjacency matrix are always zero.
- The number of 1's in each row and column equals the degree of the corresponding edge.

4.3 Definition

The edge degree d(e) of the edge e = uv is defined as the number of neighbours of e, i.e., $|N(u) \cup N(v)| - 2$.

4.4 Definition

An **algorithm** is a set of steps for accomplishing a task or solving a problem. In the context of **computer science**, an algorithm is a mathematical process for solving a problem using a finite number of steps.

5. GEC algorithm using Edge Adjacency matrix

In order to find edge chromatic number of a graph using edge adjacency matrix, a step-by-step procedure of GEC algorithm is described below.

Step 1
User Input:
Get the number of edges ‘e’ and enter the edge adjacency matrix A
Step 2
Initialization:
Initialize the color list {1,2,3……. e} for each edge
Step 3
Color Assignment:
(a)The first color is assigned for the first edge and this color is removed from the color list of all adjacent edges.
(b) For each subsequent edge the smallest available color from the color list of the associated edge is assigned and this color is removed from the color list of adjacent edges of associated edge. Repeat this step until the colors are assigned for all the edges.
Step 4
Determine Chromatic number:
The highest color value used is the edge chromatic number of the graph.
Step 5
Output:
Print the edge chromatic number and color assigned to each edge

6. Python coding for GEC algorithm

```
def edge_coloring(edge_adj_matrix):
    n = len(edge_adj_matrix)  # Number of edges
    edge_color = {}
    color_list = {i: list(range(1, n + 1)) for i in range(n)}  # Step 1: Initialize the color list
    # Step 2: Assign the first color to the first edge and remove it from adjacent edges
    edge_color[0] = color_list[0][0]
    first_color = edge_color[0]
    # Remove the assigned color from the neighboring edges
    for adj_edge in range(n):
        if edge_adj_matrix[0][adj_edge] == 1 and first_color in color_list[adj_edge]:
            color_list[adj_edge].remove(first_color)
    # Step 3: Iteratively assign colors to the rest of the edges
    for edge in range(1, n):
        # Assign the smallest available color to the current edge
        edge_color[edge] = color_list[edge][0]
```

```
        current_color = edge_color[edge]
        # Remove the assigned color from the neighboring edges' color lists
        for adj_edge in range(n):
            if edge_adj_matrix[edge][adj_edge] == 1 and current_color in color_list[adj_edge]:
                color_list[adj_edge].remove(current_color)
    # Step 4: Determine the chromatic number
    chromatic_number = max(edge_color.values())
    # Step 5: Display the chromatic number and color assignments
    print("Chromatic Number:", chromatic_number)
    for edge in edge_color:
        print(f"Edge {edge} -> Color {edge_color[edge]}")
# Function to read matrix from user input
def read_matrix():
    print("Enter the number of edges:")
    n = int(input())
    edge_adj_matrix = []
    print("Enter the edge adjacency matrix (rows separated by newline and values separated by space):")
    for _ in range(n):
        row = list(map(int, input().split()))
        edge_adj_matrix.append(row)
    return edge_adj_matrix
# Main execution
edge_adj_matrix = read_matrix()
edge_coloring(edge_adj_matrix)
```

7.Example

To understand the above algorithm for finding the edge chromatic number of any graph and to execute the python coding an appropriate example is taken and solved hereunder.

Consider the following graph with 15 edges and 10 vertices. Find the edge chromatic number of a graph with the help of the proposed GEC algorithm using python program

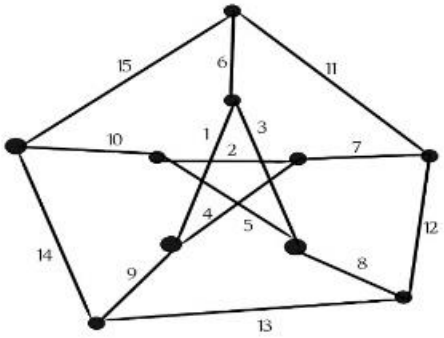

Figure:1(15,10) graph

Solution:

The screenshot of the output of the python program is given below

Output:

```
Enter the number of edges:
15
Enter the edge adjacency matrix
0 0 1 1 0 1 0 0 1 0 0 0 0 0 0
0 0 0 1 1 0 1 0 0 1 0 0 0 0 0
1 0 0 0 1 1 0 1 0 0 0 0 0 0 0
1 1 0 0 0 0 1 1 0 0 0 0 0 0 0
0 1 1 0 0 0 0 1 0 1 0 0 0 0 0
1 0 1 0 0 0 0 0 0 0 1 0 0 0 1
0 1 0 1 0 0 0 0 0 0 1 1 0 0 0
0 0 1 0 1 0 0 0 0 0 0 1 1 0 0
1 0 0 1 0 0 0 0 0 0 0 0 1 1 0
0 1 0 0 1 0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 1
0 0 0 0 0 0 1 1 0 0 1 0 1 0 0
0 0 0 0 0 0 0 1 1 0 0 1 0 1 0
0 0 0 0 0 0 0 0 1 1 0 0 1 0 1
0 0 0 0 0 1 0 0 0 1 1 0 0 1 0
Chromatic Number: 4
Edge 0 -> Color 1
Edge 1 -> Color 1
Edge 2 -> Color 2
Edge 3 -> Color 2
Edge 4 -> Color 3
Edge 5 -> Color 3
Edge 6 -> Color 3
Edge 7 -> Color 1
Edge 8 -> Color 2
Edge 9 -> Color 2
Edge 10 -> Color 1
Edge 11 -> Color 2
Edge 12 -> Color 3
Edge 13 -> Color 1
Edge 14 -> Color 4
```

Result:

By the proposed GEC algorithm, the edges of a graph is colored with minimum four colors and its chromatic number is 4.

8.Conclusion

The proposed GEC algorithm with python code helps to determine the edge chromatic number of any graph. Minimum number of lines used in python code and the given example is solved in programming

approach. The methodology used in this algorithm is simple and different from the various existing algorithm of edge coloring.

Refrences

[1] Shamim Ahmed, Applications of Graph Coloring in Modern Computer Science, January 2013, 3(2):01-07
[2] TN Sipayung, S Suwilo, P Gultom and Mardiningsih, Implementation of the greedy algorithm on graph coloring, Journal of Physics: Conference Series, Volume 2157, The 5th International Conference on Combinatorics, Graph Theory, and Network Topology (ICCGANT 2021) 21-22 August 2021, Jember, Indonesia TN Sipayung et al 2022 J. Phys.: Conf. Ser. 2157 012003 DOI 10.1088/1742-6596/2157/1/012003
[3] A. Hertz, D. de Werra, Using tabu search techniques for graph coloring, Computing 39 (1987) 345 – 351.
[4] David S. Johnson, Cecilia R. Aragon, Lyle A. Mcgeoch, Catherine Schevon, Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partitioning, Institute for operations research and the management sciences, 39(3) (1991) 378 – 406.
[5] Daniel Brelaz, New methods to color the vertices of a graph, Communications of the ACM, 22(4) (1979) 251 – 256.
[6] Amit Mittal, Parash Jain, Srushti Mathur, Preksha Bhatt, Graph coloring with minimum colors an easy approach, In the proceedings of the 2011 International conference on communication systems and network technologies, (2011) 638 – 641.
[7] K A Santoso, Dafik, I H Agustin, R M Prihandini, R Alfarisi, Vertex coloring using the adjacency matrix, In the proceedings of the 2019 IOP conference, (2019) 1 – 6.
[8] T.Ramachandran , N.Deepika, Vertex Coloring of graph using Adjacency matrix, Journal of Engineering Research and application, ISSN : 2248-9622, Vol. 10, Issue 4, (Series -V) April 2020, pp. 01-05
[9] Samsul Arifin & Co, Coloring Program of Exam Scheduling Modeling Based on Bitwise Coloring Algorithm Using Python, February 2022, Journal of Computer Science 18(1):26-32, DOI:10.3844/jcssp.2022.26.32. limited, Great Britain (1976).
[10] Ganga K. M. and Sreehari A., K. A. Germina, Python Programming for Graph Coloring Algorithms, International Journal for Research Trends and Innovation, © 2022 IJRTI | Volume 7, Issue 7 | ISSN: 2456-3315
[11] M. Saqib Nawaz, M. Fayyaz Awan, Graph coloring algorithm using adjacency matrices, International journal of scientific & engineering research, 4(4) (2013) 1840 – 1842.
[12] Charu Negi, Ajay Narayan Shukla, Vertex coloring problem approach using adjacency matrix, International journal of engineering technology science and research, 4(12) (2017) 97 – 101.
[13] Ajay Narayan Shukla, M. L. Garg, An approach to solve graph coloring problem using adjacency matrix, Bioscience biotechnology research communications, 12(2) (2019) 472 – 477.
[14] Nysret Musliu, Martin Schwengerer, Algorithm selection for the graph coloring problem,1- 15.
[15] R. M. R. Lewis, A guide to graph coloring algorithms and applications, Springer international publishing, Switzerland (2016).
[16] https://www.sciencedirect.com/science/article/abs/pii/S2214785322044108
[17] https://chatgpt.com/c/4bd9d1de-5636-4022-af5c-4c4d4ce83e6f