International Journal of Statistics and Applied Mathematics 2022; 7(2): 88-91

International Journal of Statistics and Applied Mathematics

ISSN: 2456-1452
Maths 2022; 7(2): 88-91

www.mathsjournal.com
Received: 15-12-2021
Accepted: 21-01-2022

Ayushi Malviya
Department of Mathematics, Rabindranath Tagore University, Bhopal, Madhya Pradesh, India

Bhawna Agrawal
Department of Mathematics, Rabindranath Tagore University, Bhopal, Madhya Pradesh, India

Sanjeet Kumar
Department of Mathematics Lakshmi Narain College of Technology & Science, Bhopal, Madhya Pradesh, India

Akhlak Mansuri
Department of Mathematics, Govt. Girls College, Mandsaur, Madhya Pradesh, India

Corresponding Author:
Ayushi Malviya
Department of Mathematics, Rabindranath Tagore University, Bhopal, Madhya Pradesh, India

Study of algorithm for coloring in various graph

Ayushi Malviya, Bhawna Agrawal, Sanjeet Kumar and Akhlak Mansuri

DOI: https://doi.org/10.22271/maths.2022.v7.i2b.798

Abstract
Graph coloring is an important area of mathematics and computer science. Graph coloring problem is getting more famous to solve the variety of real-world problems like map coloring, timetabling and scheduling. Graph coloring is allied with two types of coloring as vertex and edge coloring. Algorithm is a set of rules that must be followed when solving a particular coloring problem and algorithms plays an important role in graph coloring. The main objective of this paper is to study of various algorithms in graph coloring.

Keywords: Graph coloring, algorithms, graph theory

Introduction
Graph theory in mathematics is the study about the graphs. Graphs are one of the main purposes of study in discrete mathematics. In general, a graph is represented as a set of points called vertices connected by line called edges. Graph are therefore mathematical structures used to model pair wise relations between objects. They are found on maps, constellations, when constructing schemes and drawings. Graphs underlie many computer programs that make modern communication and technological cases possible. They have important role to the development of thinking, both logical and abstract. Graph theory is widely regarded as the most amusing branch of mathematics. This is because of its twin behavior; it contains the cleverest proofs in all the abstract reasoning and it has the most comprehensive range of applicability to any present-day science**.** Today graph theory has developed into a full-fledged theory from a mere collection of challenging games and interesting puzzles**.** Peculiarity of Graph theory is that it does not depends too much on other branches of Mathematics and is independent in itself. The history of graph theory may be particularly detected to 1735, when the Swiss mathematician Leonhard Euler gave the solution of the Königsberg bridge problem (Thang Bui N *et al*. 2008) [1].
The Königsberg bridge problem was an old puzzle regarding the possibility of searching a path over every one of seven bridges that span a forked river flowing past an island-but without using any bridge twice. Euler argued that no such path exists. His proof include only references to the physical arrangement of the bridges, but basically, he proved the first theorem in graph theory (Aslan M *et al*. 2016) [2].

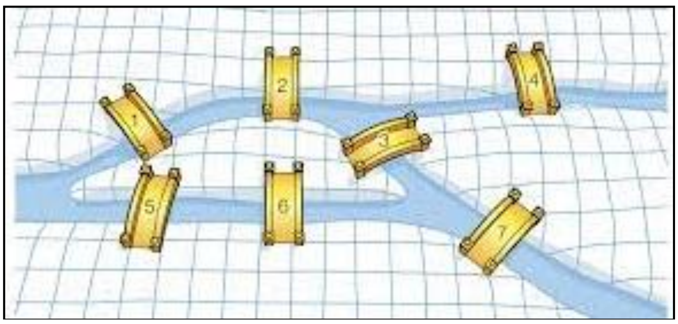

Fig 1: Bridges of Konigsberg

Graph Coloring

As used in graph theory Graph coloring problem is getting more popular to solve the problem of coloring the adjacent regions in a map with minimum different number of colors. It is used to solve a variety of real-world problems such as map coloring, timetabling and scheduling. Graph coloring is allied with two types of coloring as vertex and edge coloring. The aime of the both types of coloring is to color the whole graph without conflicts. Therefore, adjacent nodes or adjacent edges must be colored with different colors. The number of the minimum possible colors to be used for Graph coloring problem is called chromatic number. As the number of nodes or edges in a graph increases, the difficulty of the problem also increases. Because of this, each algorithm cannot discover the chromatic number of the problems and may also be different in their executing times. Due to these constructions, Graph coloring problem is known an NP-hard problem. Various heuristic and Meta heuristic methods have been developed in order to solve the graph coloring problem (Jyothi VJ *et al*. 2020) [3].

Vertex-Colorings and Edge-Colorings

Given a set C called the set of colors (these could be numbers, letters, names, whatever), a function which assigns a value in C to each vertex of a graph is said to be a vertex-coloring. An ideal vertex-coloring never assigns adjacent vertices the same color. Similarly, a function which assigns a value from a set of colors C to each edge in a graph is said to be an edge-coloring (Dr. Omari *et al*. 2006) [4].

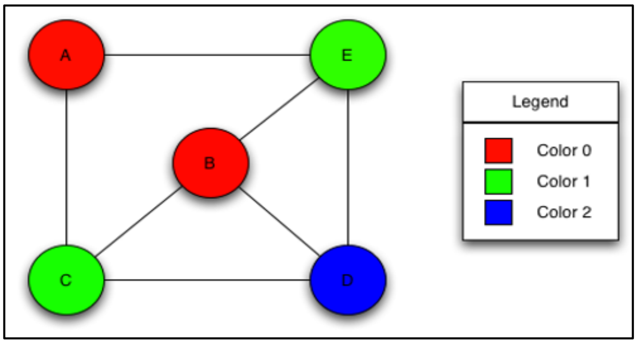

Fig 2: Vertex-Coloring Graph

In its simplest form Graph Coloring, is a way of coloring the vertices of a graph such that two adjacent vertices does not share the same color, said a vertex coloring. Similarly, an edge coloring assigns a color to each edge so that two incident edges does not share the same color (Subarna Sinha *et al*. 2014) [5].

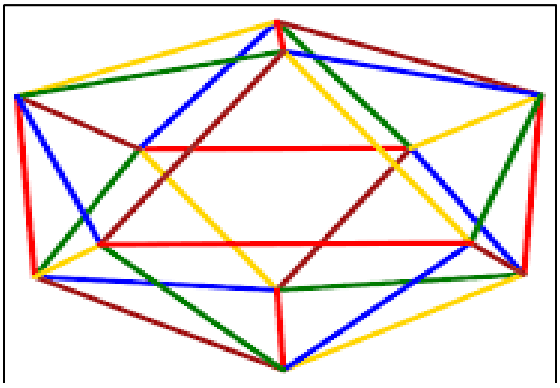

Fig 3: Edge-Coloring Graph

Region Coloring: Region coloring is an assigning, colors to the regions of a planar graph such that two adjacent regions does not have the same color. Two regions are called adjacent if they have a same edge (Celia Glass A *et al*. 2003) [6].

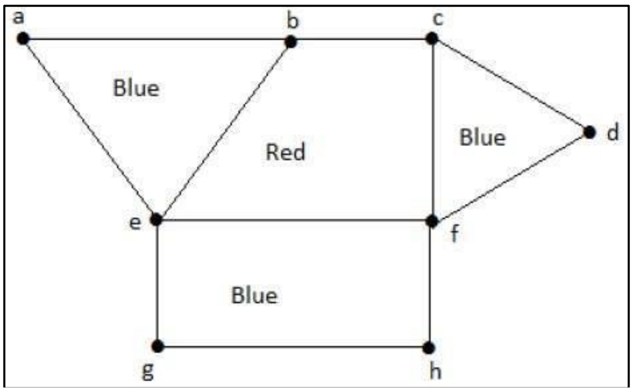

Fig 4: Region colouring Graph

K-colorable

A k-coloring of a graph G, is a labeling of the vertices f: V (G) → S, where S is some set such that cardinality of S is k. Normally the set S as a collection of k different colors, say S = {green, blue, red, yellow etc.}, or more abstractly as the non negative integers S = {1, 2, k}. The labels are called colors. A k-coloring is proper if adjacent vertices are different colors. A graph is said to be k-colorable if it has a proper k-coloring. The chromatic number χ (G) is the minimum positive integer k such that G is k-colorable. A graph is said to be k-colorable if and only if it is k-partite. In other words, k-colorable and k-partite mean the same thing. Determining the k different partite sets of a k-colorable graph and conversely determine a k-coloring of a k-partite graph. In general it is not easy to find the chromatic number of a graph or even if a graph is k-colorable for a given k (Philippe Galinier *et al*. 1999) [7].

Graph coloring algorithms

The word algorithm means" a set of rules or process to be comply in calculation or other solving problems" therefore algorithm refers to a set of instructions or rules that show haw step by step perform a task to be executed upon in order to get the expected results (Diaz IM *et al*. 2005) [8].

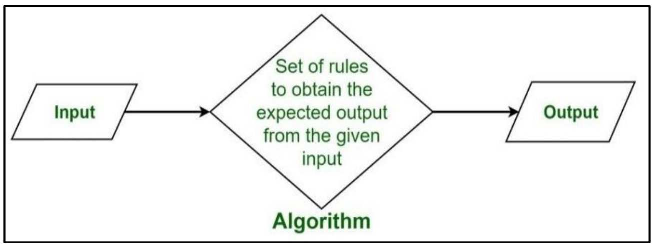

Fig 5: Graph colouring algorithms

Greedy coloring

The greedy algorithm considers the vertices in a specific order $v_1 \ldots\ldots v_n$ and assigns to v_1, the smallest available color not used by u_i's neigh bours among v_1....-1, adding a fresh color if required. The quality of the resulting coloring depends on the elected ordering. There exists an ordering that leads to a greedy coloring with the optimum number of x (G) colors. On the other side, greedy colorings can be arbitrarily bad; for example, the crown graph on n vertices can be tow-colored, but has an ordering that leads to a greedy coloring with n/2 colors. For chordal graphs, and for specific cases of chordal graphs the greedy coloring algorithm can be used to discover optimal colorings in polynomial time, by selecting the vertex ordering to be the reverse of a perfect deletion ordering for the graph. The completely orderable graphs generalize this property, but in NP-hard to discover a perfect ordering of these graphs. Many other graph coloring heuristics are similarly based on greedy coloring for a specific static or dynamic design of ordering the vertices, these algorithms are also called sequential coloring algorithms (Halawi M *et al*. 2018) [9].

The maximum number of colors that can be obtained by the greedy algorithm, by using a vertex ordering chosen to maximize this number is called the Grundy number of a graph (Yeşil Ç *et al*. 2011) [10].

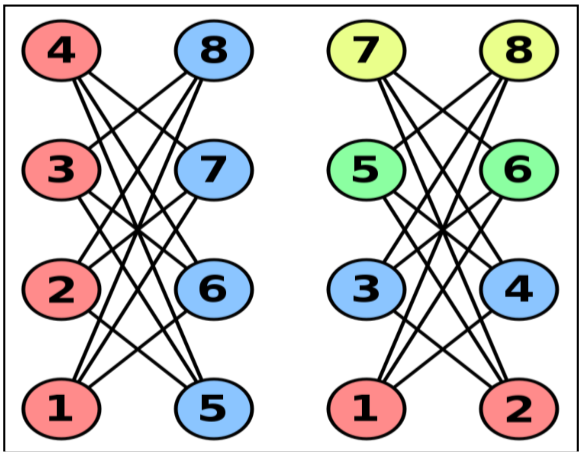

Fig 6: Bipartite Graph Colour

Determining if a graph can be colored with 2 colors is equivalent to determining whether or not the graph is bipartite, and thus computable in linear time using breadth-first search or depth-first search. More generally, the chromatic number and a corresponding coloring of perfect graphs can be computed in polynomial time using semi definite programming. Closed formulas for chromatic polynomial are known for many classes of graphs, such as forests, chordal graphs, cycles, wheels, and ladders, so these can be evaluated in polynomial time (Thomson Leighton 2011) [11].

Decentralized algorithms

Decentralized algorithms are ones where no message passing is allowed (in contrast to distributed algorithms where local message passing takes places), and efficient decentralized algorithms exist that will color a graph if a proper coloring exists. These assume that a vertex is able to sense whether any of its neighbors are using the same color as the vertex i.e., whether a local conflict exists. This is a mild assumption in many applications e.g. in wireless channel allocation it is usually reasonable to assume that a station will be able to detect whether other interfering transmitters are using the same channel (e.g. by measuring the SINR). This sensing information is sufficient to allow algorithms based on learning automata to find a proper graph coloring with probability one (Haidar Harmanani *et al*. 2006) [12].

Recursive Largest First (RLF)

RLF algorithm is one of the most famous greedy heuristics for the vertex coloring problem. It sequentially constructs color classes on the basis of greedy choices. In particular the first vertex fixed in a color class C is one with a maximum number of uncolored neighbors, and the next vertices fixed in C are chosen so that they have as many uncolored neighbors which cannot be placed in C. These type of greedy selections can have a significant effect on the performance of the algorithm Polynomial time (Ganguli R, 2017; Poddar N, 2018) [13, 14].

Dsatur Algorithm

The Dsatur algorithm is similar to the Greedy algorithm in that once the vertex has been selected, it is assigned to the lowest color lable not assigned to any of its provide the main power behind the algorithm in that they prioritise vertices that are seen to be the “most constrained” - that is, vertices that currently have the fewest Colour options available to them. Consequently, these “more constrained” vertices are dealt with first, allowing less constrained vertices to be colored later. Because the Dsatur algorithm generates a vertex ordering during execution, the number of colors it uses is more predictable than the greedy algorithm. Its solutions also tend to have fewer colors than those of the greedy algorithm. One feature of the algorithm is that, if a graph is composed of multiple components, then all vertices of a single component will be colored before the other vertices are considered. Dsatur is also accurate for several graph topologies including bipartite graphs, cycle graphs and wheel graphs (Borodin OV, 2005; Islam T, 2016) [15, 16].

Genetic algorithm

Genetic algorithm is a calculation model simulating natural selection in the Darwinian biological evolution and biological evolution in genetics. Before solving the problem, we are supposed to conduct mathematical modeling first. A possible solution is seen as a chromosome. Possible solutions are always made up of multiple elements and each element is called a gene. As for the circulating of the algorithm, at first, it will randomly generate a set of possible solutions, the first generation of chromosomes, called a population. After that, the fitness function, which is used to develop the fitness of these chromosomes will calculate the fitness of each chromosome and the probability that each chromosome will be selected in the next development according to the fitness. Next, crossover will happen. Crossover means seeking two parent chromosomes, cutting them at the same location and join together, thus forming a new chromosome. The new chromosome consist some genes from the father chromosome and some genes from the mother chromosome. By crossover, n-m pieces of new chromosome will get arise. After crossover, variation will take place on the new-arise n-m pieces of new chromosome. Variation is to select best genes by swapping the combination order to break the current search limit and the algorithm can find out the best solution. At last, m pieces of new chromosomes will get arise by copying, which means the chromosomes with the highest fitness of the previous generation are directly copied intact to the next generation. Assuming that n chromosomes need to be generated in each development, n-m chromosomes need to be generated in each evolution by crossover, and the remaining m chromosomes are generated by copying m chromosomes with the highest quality of the previous generation (Islam T, 2016; Shukla AN, 2019) [17, 18].

Conclusion

Graph coloring is still a very active field of research, we have studied and presented introduction of RLF algorithm, Greedy algorithm, Dastur algorithm, Genetic algorithm, Decentralized algorithms, for graph coloring problem.

References

1. Thang Bui N, Thanh Vu H, Nguyena Chirag M, Patel, Kim-Anh Phan T. An ant-based algorithm for graph coloring, Discrete Applied Mathematics. 2008;156:190-200.
2. Aslan M, Baykan NA. A performance comparison of graph coloring algorithms. International Journal of Intelligent Systems and Applications in Engineering. 2016;4(1)1-7.
3. Jyothi VJ, Kumar S. A polynomial-time algorithm for minimal roman coloring of graph. International Journal for Research in Applied Science & Engineering Technology. 2020;8:106-111.
4. Dr. Omari, Eddin Sabri. New Heuristic graph-coloring algorithms. American Journal of Mathematics and Statistics. 2006;2(4):739-741.
5. Subarna Sinha, Suman Deb. Graph Coloring Problem Solution Using Modified Flocking Algorithm. Article *in* Advances in Intelligent Systems and Computing. 2014;259:113-123.
6. Celia Glass A, Adam Prugel-Bennett. Genetic Algorithm for Graph. Journal of Combinatorial Optimization. 2003;4(3):379-397.
7. Philippe Galinier, Jin-Kao Hao. Hybrid Evolutionary Algorithms. Journal of Combinatorial Optimization. 1999;3(4):379-397.
8. Diaz IM, Zabala Z. A Branch-and-Cut algorithm for graph coloring, free graphs and similar graph classes. Discrete Applied Mathematics. 2005;154(5):826-847.
9. Halawi M, Ramzi AH. Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem. Procedia Computer Science. 2018;126:899-906.
10. Yeşil Ç, Yılmaz B, Korkmaz EE. Hybrid local search algorithms on Graph Coloring Problem. International Conference on Hybrid Intelligent Systems (HIS); c2011. p. 468-473.
11. Thomson Leighton. A Graph Coloring Algorithm for Large Scheduling. Journal of Research of the Notional Bureau of Standards. 1979;84:6.
12. Haidar Harmanani, Hani Abas. A method for the minimum coloring problem using genetic algorithm Conference. The 17th IASTED International Conference on Modeling and Simulation; 2006.
13. Ganguli R, Roy S. A Study on Course Timetable Scheduling using Graph Coloring Approach. International Journal of Computational and Applied Mathematics. 2017;12(2):469-485.
14. Poddar N, Mondal B. An instruction on course timetable scheduling applying graph coloring approach. International Journal of Recent Scientific Research Research. 2018;9(3):23939-23945.
15. Borodin OV, Glebov AN, Raspaud A, Salavatipour MR. Planar graphs without cycles of length from 4 to 7 are 3-colorable. Journal of Combinatorial Theory. 2005;93(2):303-311.
16. Islam T, Mohammad Z, Perves A, Hasan M. University Timetable Generator Using Tabu Search. Journal of Computer and Communications. 2016;4(16):28-37.
17. Shukla AN, Bharti V, Garag ML. A linked list-based exact algorithm for graph coloring problem. Revue d'Intelligence Artificielle. 2019; 33(3):189-195.
18. Bouzenada M, Saidou ED. A Generalized Graph Strict Strong Coloring Algorithm. International Journal of Applied Metaheuristic Computing. 2012;3(1):24-33.