

APPLICATIONS OF GRAPH THEORY IN COMPUTER SCIENCE

* *Dr. Smt. Megha Abhiman Bhamare*, Assistant Professor of Mathematics, K. V. N. Naik College, Nasik

I. INTRODUCTION

Graph theory is a branch of mathematics concerned with networks of points connected by lines. The subject of graph theory had its beginning in recreational math problems, but it has grown into a significant area of mathematical research with applications. The origin of graph theory can be traced back to Euler's work on the Konigsberg bridges problem (1735), which subsequently led to the concept of an Eulerian graph. The study of cycles on polyhedral by the Thomas P. Kirkman (1806-1895) and William R. Hamilton (1805- 1865) led to the concept of a Hamiltonian graph. The concept of a tree without cycles, appeared implicitly in the work of Gustav Kirchhoff (1824-1887), who employed graph- theoretical ideas in the calculations of currents in electrical networks or circuits. Later, Arthur Cayley (1821-1895), Jame J. Sylvester (1806-1897) George Polya (1887-1885) and others use 'tree' to enumerate chemical molecules. The study of planer graphs originated in two recreational problems involving the complete graph K_5 and the complete bipartite graph $K_{3,3}$. These graphs prove to be planarity, as was subsequently demonstrated by Kuratowski. First problem was presented by A. F. Mobius around the year 1840. Here the problem is that deciding whether the graph K_5 is planer. The origin of second problem is unknown but it is first mentioned by H. Dudeney in 1913 in its present form. The puzzle is to lay a water, gas and electricity to each of the three houses without any pipe crossing another. This problem is that deciding whether the graph $K_{3,3}$ is planer. The celebrated four-color problem was first posed by Francis Guthrie in 1852 and a celebrated incorrect "proof" by appeared in 1897 by Alfred B. Kempe. It was proved by Kenneth Appel and Wolfgang Haken in 1976 and a simpler and more systematic proof was produced by Neil Robertson, Daniel Sanders, Paul Seymour and Robin Thomas in 1994.

II. GRAPHS & GRAPH THEORY

In Theory, a graph bears no relation to the graphs that chart data, such as the progress of the stock market or the growing population of the planet. Graph paper is not particularly useful for drawing the graphs of Graph Theory. In Mathematics and computer science, graph theory is the study of graphs which are mathematical structures used to model pair wise relations between objects. A "graph" in this context is made up of "vertices" or "nodes" and lines called edges that connect a graph. A graph may be undirected meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another and for other variations in the types of graph that are commonly considered.

III. APPLICATIONS OF GRAPH THEORY

Graph Theory in Operations Research

Graph theory is a very natural and powerful tool in combinatorial operations research. Some important operations research problems that can be solved using graphs. A network called transport network where a graph is used to model the transportation of commodity from one place to another. The objective is to maximize the flow or minimize the cost within the prescribed flow. The graph theoretic approach is found more efficient for these types of problems though they have more constraints.

Graph Coloring

Graph coloring is one of the most important concepts in graph theory and is used in many real time applications in computer science. Various coloring methods are available and can be used

on requirement basis. The proper coloring of a graph is the coloring of the vertices and edges with minimal number of colors such that no two vertices should have the same color. The minimum number of colors is called as the chromatic number and the graph is called properly colored graph.

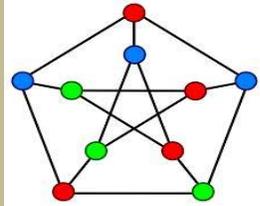


Figure-1
Proper vertex coloring with Chromatic number 3

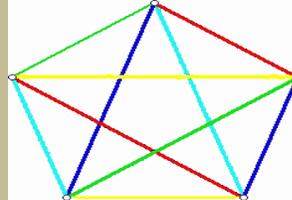


Figure-2
Proper vertex coloring with Chromatic number 5

Graph coloring techniques in scheduling

Here some scheduling problems that uses variants of graph coloring methodologies such as precoloring, list coloring, multicoloring, minimum sum coloring.

Job scheduling

Here the jobs are assumed as the vertices of the graph and there is an edge between two jobs if they cannot be executed simultaneously. There is a 1-1 correspondence between the feasible scheduling of the jobs and the colorings of the graph.

Aircraft scheduling

Assuming that there are k aircrafts and they have to be assigned n flights. The i^{th} flight should be during the time interval (a_i, b_i) . If two flights overlap, then the same aircraft cannot be assigned to both the flights. This problem is modeled as a graph as the vertices of the graph correspond to the flights. Two vertices will be connected, if the corresponding time intervals overlap. Therefore, the graph is an interval graph that can be colored optimally in polynomial time.

Bi-processor tasks

Assume that there is a set of processors and set of tasks. Each task has to be executed on two processors simultaneously and these two processors must be pre assigned to the task. A processor cannot work on two jobs simultaneously. This type of tasks will arise when scheduling of file transfers between processors or in case of mutual diagnostic besting of processors. This can be modeled by considering a graph whose vertices correspond to the processes and if there is any task that has to be executed on processors i and j , then an edge to be added between the two vertices. Now the scheduling problem is to assign colors to edges in such a way that every color appears at most once at a vertex. If there are no multiple edges in the graph i.e. no two tasks require the same two processors then the edge coloring technique can be adopted. The authors have developed an algorithm for multiple edges which gives a 1-1 approximate solution.



Figure-3: Tasks allocated to processors P1, P2 P3, P4, P5, P6 & P7

The diagram shows the tasks namely task1, task2, task3 and task4 are allocated to the processors (P1, P5); (P1, P6); (P2, P4) and (P3, P7) respectively

Multicoloring

In the multicoloring problem each vertex v has a demand $x(v)$, and we have to assign a set of $x(v)$ colors to each vertex v such that neighbors receive disjoint sets of colors. Multicoloring can be used to model the scheduling of jobs with different time requirements: the set of colors assigned to vertex v corresponds to the $x(v)$ time slots when we work on the job. There are two main variants of multicoloring. In non-preemptive multicoloring the set of colors assigned to a vertex has to be a continuous interval of colors. This reflects the requirement that the jobs cannot be interrupted, they have to receive a continuous time window. On the other hand, in preemptive multicoloring we assume that the jobs can be interrupted, hence the set of colors assigned to a vertex can be arbitrary, it does not have to be continuous.

Precoloring extension

In certain scheduling problems we do not have full control over the schedule, the assignments of certain jobs are already decided. In this case some of the vertices of the conflict graph has a preassigned color, and we have to solve the precoloring extension problem: extended the coloring of these vertices to the whole graph, using the minimum number of colors. Biro, Hujter and Tuza started a systematic study of precoloring extension. There is a maintenance period for each aircraft, during which it cannot fly. We can model these maintenance periods by adding a “dummy” flight for the maintenance period of each aircraft and requiring that the maintenance period of the i^{th} aircraft is assigned to the i^{th} aircraft. Therefore we have to solve the precoloring extension problem on the conflict graph, which is an interval graph. The precoloring extension problem is NP-complete for interval graphs, but it can be solved in polynomial time if every color is used only once in the precoloring, that is, if every aircraft has only one maintenance interval.

List coloring

In the list coloring problem each vertex v has a list of available colors and we have to find a coloring where the color of each vertex is taken from its list of available colors. List coloring can be used to model situations where a job can be processed only in certain time slots or if it can be processed only by certain machines. Using standard dynamic programming techniques, list coloring can be solved in polynomial time on trees and partial k -trees. By combining dynamic programming with a clever use matching, list coloring can be solved on the edges of trees as well. The multicoloring concept can be applied for list colorings as well: each vertex has an integer demand $x(v)$, and vertex v has to receive a set of $x(v)$ colors from its list of colors. The algorithm for list coloring trees and partial k -trees does not generalize for the multicoloring case, as the problem is NP-complete already for binary trees. On the other hand, list edge multicoloring can be solved in polynomial time on trees: using standard techniques, the good characterization theorem of Marcotte and Seymour can be turned into a polynomial time algorithm. This result is generalized into a slightly more general class of graphs, that includes odd cycles. Moreover, a randomized algorithm is given for an even more general class of graphs, including even cycles.

Map coloring and GSM mobile phone networks

Groups Special Mobile (GSM) is a mobile phone network where the geographical area of this network is divided into hexagonal regions or cells. Each cell has a communication tower which connects with mobile phones within the cell. All mobile phones connect to the GSM network by searching for cells in the neighbors. Since GSM operate only in four different frequency ranges, it is clear by the concept of graph theory that only four colors can be used to color the cellular

regions. These four different colors are used for proper coloring of the regions. Therefore, the vertex coloring algorithm may be used to assign at most four different frequencies for any GSM mobile phone network i.e. if a given a map drawn on the plane or on the surface of a sphere, the four color theorem asserts that it is always possible to color the regions of a map properly using at most four distinct colors such that no two adjacent regions are assigned the same color. Now, a dual graph is constructed by putting a vertex inside each region of the map and connect two distinct vertices by an edge if and only if their respective regions share a whole segment of their boundaries in common. Then proper coloring of the dual graph gives proper coloring of the original map. Since, coloring the regions of a planar graph G is equivalent to coloring the vertices of its dual graph and vice versa. By coloring the map regions using four color theorem, the four frequencies can be assigned to the regions accordingly.



Figure-4: Map coloring of the states in the U.S

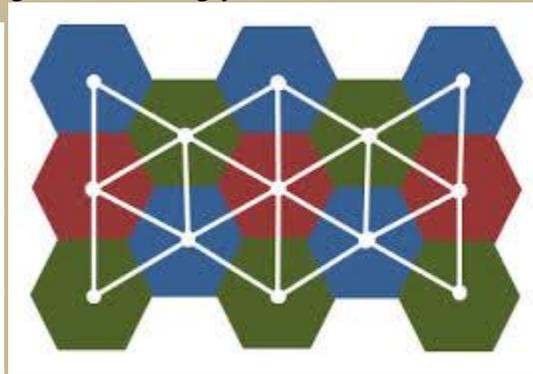


Figure-5: The cells of a GSM mobile phone network

Algorithms and graph theory

The major role of graph theory in computer applications is the development of graph algorithms. Numerous algorithms are used to solve problems that are modeled in the form of graphs. These algorithms are used to solve the graph theoretical concepts which intern used to solve the corresponding computer science application problems. Some algorithms are as Shortest path algorithm in a network, Finding a minimum spanning tree ,Finding graph planarity , Algorithms to find adjacency matrices, Algorithms to find the connectedness , Algorithms to find the cycles in a graph , Algorithms for searching an element in a data structure and so on. Various computer languages are used to support the graph theory concepts. The main goal of such languages is to enable the user to formulate operations on graphs in a compact and natural manner Some graph theoretic languages are SPANTREE (to find a spanning tree in the given graph), GPL (Graph Theoretic Language) ,GASP (Graph Algorithm Software Package), HINT (extension of LISP), GRASPE(extension of LISP), IGTS (extension of FORTRAN), GEA (Graphic Extended) ALGOL (Extension of ALGOL), AMBIT(to manipulate digraphs), GIRL (Graph Information Retrieval Language) and FGRAAL (FORTRAN Extended Graph Algorithmic Language).

Graph algorithm in computer network security

The vertex cover algorithm (Given as input a simple graph G with n vertices labeled $1, 2, \dots, n$, search for a vertex cover of size at most k . At each stage, if the vertex cover obtained has size at most k , then stop.) is used to simulate the propagation of stealth worms on large computer networks and design optimal strategies to protect the network against virus attacks in real time. Simulation was carried out in large internet like virtual network and showed that the topology routing has big impact on worm propagation. The importance of finding the worm propagation is to hinder them in real time. The main idea here is to find a minimum vertex cover in the graph whose vertices are the routing servers and the edges are the connections between the routing

servers. Then an optimal solution is found for worm propagation and a network defense strategy is defined.

Graph theory relevant to ad-hoc networks

In Ad-hoc networks, issues such as connectivity, scalability, routing, modeling the network and simulation are to be considered. Since a network can be modeled as a graph, the model can be used to analyze these issues. Graphs can be algebraically represented as matrices. Also, networks can be automated by means of algorithms. The issues such as node density, mobility among the nodes, link formation between the nodes and packet routing have to be simulated. To simulate these concepts random graph theory is used. The connectivity issues are analyzed by using graph spanners, (A geometric spanner or a k-spanner graph or a k-spanner was initially introduced as a weighted graph over a set of points as its vertices and every pair of vertices has a path between them of weight at most k times the spatial distance between these points, for a fixed k.) proximity graphs, (A proximity graph is simply a graph in which two vertices are connected by an edge if and only if the vertices satisfy particular geometric requirements), sparsification and spectral graph theory. Various algorithms are also available to analyze the congestion in MANET's where these networks are modeled based on graph theoretical ideas.

IV. CONCLUSION

This paper focused on the various applications of major graph theory that have relevance to the field of computer science. The main aim of this paper is to present the importance of graph theoretical ideas in various areas of compute applications for researches that they can use graph theoretical concepts for the research.

References

1. Narasingh Deo (1990), Graph theory with applications to engineering and computer science, Prentice Hall of India.
2. E. G. Coffman, Jr., M. R. Garey, D. S. Johnson, and A. S. LaPaugh, (1985), Scheduling file transfers, SIAM J. Comput., 14(3):744–780
3. A. Graf, M. Stumpf, and G. Weißenfels (1998), On coloring unit disk graphs, *Algorithmica*, 20(3):277–293.
4. J. A. Hoogeveen, S. L. van de Velde, and B. Veltman, (1994), Complexity of scheduling multiprocessor tasks with prespecified processor allocations, *Discrete Appl. Math.*, 55(3):259–272.
5. I. Holyer, (1991), The NP-completeness of edge-coloring, *SIAM J. Comput.*, 10(4):718–720.
6. Shariefuddin Pirzada and Ashay Dharwadker, (2007), *Journal of the Korean Society for Industrial and applied Mathematics*, Volume 11, No.4.
7. K. Appel and W. Haken (1976), Every Planar Map is Four Colorable, *Bull. Amer. Math. Soc.* 82 711-712.
8. Ashay Dharwadker, (2006), The Vertex Coloring Algorithm, http://www.dharwadker.org/vertex_coloring
9. Ashay Dharwadker (2000), A New Proof of The Four Colour Theorem, <http://www.dharwadker.org>
10. Shariefuddin Pirzada and Ashay Dharwadker (2007), *Graph Theory*, Orient Longman and Universities Press of India.
11. Daniel Marx, “Graph Coloring problems and their applications in scheduling”,

12. <http://www.britinaca.com/bps/additionalcontent/18/33373769/concepts-of-graph-theory-relevant-to-Adhoc-Networks>
13. http://en.wikipedia.org/wiki/Geometric_spannerW