Journal of Advances in Science and Technology

Vol. V, Issue No. IX, May-2013, ISSN 2230-9659

A STUDY ON INTERSECTION GRAPHS

AN INTERNATIONALLY INDEXED PEER REVIEWED & REFEREED JOURNAL

www.ignited.in

Journal of Advances in Science and Technology
Vol. V, Issue No. IX, May-2013, ISSN 2230-9659

A Study on Intersection Graphs

Dr. Sarita Devi

Asst. Professor. Indira Gandhi National College Ladwa Kurukshetra

Abstract – The study of intersection graphs was started in the early 1950's. Chordal graphs are very important type of intersection graphs. A major investigation of chordal graph: under the frame work of intersection graphs was done by Gavril. It was Gavril who proved the very important result that chordal graphs are intersection graphs of subtrees of a tree. Interval graphs and circular arc graphs are very important types of subtree graphs and wert: studied by Gilmore and Hoffman.

----------♦----------

INTRODUCTION

A directed Hamilton path of ***D*** is a directed path that includes every vertex of D. It is known that, every tournament has a directed Hamilton path. A directed cycle is a directed tour of length k *z* 1 in which vo = vk and vertices vo, vl, ..., vk.1 are distinct. **A** directed Hamilton cycle of D is **a** directed cycle that includes every vertex of D.

A directed acyclic graph is a digraph D with no cycles. Let u, v be the vertices in V (D). The vertex u is an ***ancestor*** of v; and v is a ***descendant*** of u, if there is a uv - directed path ill D; otherwise u and v are unrelated. If u is an ancestor of v and(uy)is an arc, then u is ***a parent*** of v, and v is a ***child*** of u. A ***rooted tree*** is a directed acyclic graph in which all vertices have ***indegree*** one, and a specially designated node, called the ***root,*** has indegree zero. The ***root*** of a rooted tree T is denoted by ***root*** *(T).* The subtree of T rooted at v is the subtree of T induced by the descendants of v.

The ***in-degree*** d-~(vo)f ***a*** vertex v in D is the number of arcs with head v; and ***out-degree*** d+~(vo)f v is the number of arcs with tail v. We denote the minimum and maximum in-degrr:es and out-degrees of D by **6-** (D), **A-** (D) and **6'** (D) and **A'(D)** respectively. Also, a directed graph ***G*** (V, E) is rooted at vertex r if there is a path from r to every vertex in V. G(V, E) is also called a rooted directed acyclic graph with root r.

A subset A V of r ve~tic es is an ***r-clique*** if it induces a complete subgraph. That is, if A *c* V is an r-clique, then G **[A]** E K,. A single vertex is a 1-clique. A clique A is ***maximum,*** if there is no clique of G which properly contains **A.** A clique is ***maximum*** if there is no clique of G of larger cardinality. The ***clique number*** to(G) is the number of vertices in a maximum clique of G. A ***clique cover*** of size k is a partition of the vertices, V = A, + **A2 +...Ak,** such that each **A,** is a clique. The ***clique cover number*** of G, k(G) is the size of a smallest possible cover of G.

An ***independent set*** or a stable set of a graph G is a subset X of vertices, where no two of which are adjacent. The ***stability number*** of G, a(G), is the number of vertices in a stable set maximum cardinality.

A ***proper c-coloring*** is a partition of the vertices V = XI + X2 + ... + **X,** such that each Xi is a stable set. Here the members of Xi are painted with the color i, and adjacent vertices will receive different colors. The ***chromatic number*** of G, x(G), is the smallest possible number c, for which there exists a proper c - coloring of G.

Corresponding to the above properties of a graph G, we have the following results; o(G) < x(G) and a(G) < k(G). Since every vertex of a maximum clique (maximum stable set) must be contained in a different partition segment in any minimum proper coloring (minimum clique cover), we get, o(G) = a(G) and x(G) = k(G). A set S is an independent set if G[S] is a null graph. For any graph, the intersection of a clique and a stable set can be at most one vertex.

Also, for any graph, G, o(G) **I** ;1,(G) and a(G) **I** k (G). These inequalities are dual to one another since, a(G;t = w(GC) and k(G) = x(GC). Cliques and Independent sets have been studied by Fred Galvin [Gal 20001. A graph G is a comparability graph if and only if each odd cycle has at least one triangular chord .

REVIEW OF LITERATURE:

Antonysamy identified two classes of intersection graphs depending upon the nature of the vertex paths. They are the subclasses of vertex path graphs, the CV graphs and the PV graphs. A family of paths in a tree T is said to be ***Compact,*** if no edge of T is in more than one path of the family. The intersection graph of a family of Compact vertex paths in a tree is called a ***Compact Vertex path graph*** or ***(CVgraph.*** Also, a family of paths in a tree T is said to ***be perfect,*** if no vertex of T is an internal vertex in more than one path of the family. The intersection grsph of

a family of Perfect vertex paths in a tree is called a ***Perfect Vertex Path graph*** of ***PV Graph.*** Various properties of PV and CV graphs have been discussed in [An 941] and recognition algorithms for both these intersection graphs have been introduced. The vertex disjoint paths in a tree is further studied by Pancia and Mohanty [PM 95] and a polynomial time recognition algorithm was also presented.

The Interval graphs are intersection polysemic with strongly chordal graphs.

Proof

By Farber [Fa **83],** interval graphs are strongly chordal graphs. We have proved in CH 111 that the interval gr;iphs are intersection polysemic with interval graphs in the same vertex set. 1Ience the interval graphs are intersection polysemic with strongly chordal graphs.

METHODOLOGY:

A multigraph is ***a*** graph with multiple edges between pairs of vertices. ***A line graph*** L(M) of a multigrai~h M has a vertex for every edge of M with two vertices adjacent in L(M) exlctly when the corresponding edges in M are adjacent When H = L(M) we say that M = L-'(H). **A** multigraph M is bipartite if the vertices can be partitioned into two parts so that no two vertices in the same part are adjacent. A multigraph M is ***triangle free*** if it does not contain three mutually adjacent vertices. ***A star tree*** is a connected bipartite graph G with one part consisting of a single vertex.

Matrix representation of a graph

Adjacency Matrices, Claque Matrices etc. can be used to represent a graph. A weighted graph is represented by the ***weight matrix*** whose elements are the weights of the edges. Let G (V, E) be a graph whose vertices have been ordered arbitrarily as vl, vz, ..., v,. Then the adjacency matrix M = (m,,) of G is an n x n matrix with entries **m,,** = 0, if (v,,vJ) **B.** E, and m,J = I, if (v,, v,) **E** E. The maximal cliques - verses - vertices incidence matrix of an undirected graph is called a ***clique matrix,*** if all the maximal cliques are included.

Triangulated Graph

Every simple cycle of length strictly greater than **3** possess a chord. This property is called the ***triangulated graph property.*** Graphs which satisfy this property are called ***triangulated graphs.*** One of the first classes of graphs to be recognized as being perfect was the class of triangulated graph.

The interval graph constitute a special type of triangulated graph. Triangulated graphs have also been called ***Chordal, Rigid Circuit, Monotone Transitive, and Perfect Elimina,ion Graphs.*** A vertex v of G is called ***simplicial*** if its adjacency set Adj(v) induces a complete subgraph of G; i.e.; ~idj(v) is a clique (not necessarily maximal).

Every triangulated graph G (V, E) has a simplified vertex. Moreover, if G is not a clique, then it has two nonadjacent simplified vertices**.** Triangulated graphs can be recognized in linear time. **A** graph is triangulated, if and only if it is the intersection graph of a family of sub-trees of a tree.

Interval Graphs

An undirected graph G is called an ***interval graph*** if its vertices can be put into one-to-one correspondence with a set of intervals I of a linearly ordered set such that two vertices are connected by an edge of G if and only if their corresponding intervals overlap at least partially or, have nonempty intersection.

If these intervals are of unit length, we get a ***unit interval graph.*** If no interval properly contains another, the graph constructed from that family of intervals on a line will be called ***improper interval graph.*** It is well known that the classes of unit interval graphs and proper interval graphs coincide.

A graph G is an interval graph if and only if it is chordal and contains no asteroidal triples. The coloring, clique, stable set, and clique cover problems can be solved in polynomial time for i~iterval graphs. The earliest characterization of interval graphs was obtained y Lekkerkerker and Boland**.** Their result embodies the notion that an int1:rval graph neither branch into more than two directions, nor circle hack into itself.

Block Graphs

A cut point of a graph is one whose removal increases the number of components of the graph. Thus if v is a cut point of a connected graph G, then G -v is disconnected. **A** non separable graph is connected, nontrivial and has no cut points. ***A block*** of a graph is a maximal non separable subgraph.

If we take the blocks of G as the family of sets, then the intersection graph ***C2*** (F) is the ***block graph*** of G, denoted by B(G). The blocks of G correspond to the points of **I)** (G) and two of these points are adjacent whenever the corresponding blocks contain a common cut point of G. Clearly block graphs are subclass of chordal graphs. **Hanary** proved that, a graph H is the block graph of some graph if and only if every block of H is complete.

A path graph is the intersection graph of paths in a tree; taking the paths to be the set of edges making up the path we get the ***Edge Path Graphs*** and taking the paths to be the set of vertices making up the path.

Efficient algorithm : Fast algorithm

An algorithm is ***efficient or fast*** if its complexity is a polynomial in the input size n, of the data.

A computational problem is called ***tractable*** if there exists an efficient algorithm for solving the problem. **A** computational problem is ***intractable*** if it can be established that then is no algorithm to solve the problem. Or, a problem is intractable if all algorithms to solve the problem are of at least exponential time complexity. The implication of this rating scheme is that problems having polynomial time bounded algorithms are tractable.

Search Algorithms

Given an alphabetical listing of words, we wish to determine if and where a specific word appear: on the list. A simple way is to start at the beginning of the list and to search in sequence until the name is found or until we come to the end of the list. This algorithm is often called ***sequential search algorithm.*** If there are n words on the list and if we want to know the position of a particular word then it will take n comparisons before the algorithm terminates. Hence the complexity is measured in terms of the number of comparisons rather than directly as time units. Then the ***worst case complexity of this algorithm is O(n).*** For a sequential search algorithm it is not necessary that the words have to be in alphabetical order.

Sorting Algorithms

In the search algorithms we assume that the given list is in alphabetical order but the sorting algorithms are useful to arrange a given list of words in alphabetical order. There are two algorithms to do this job, one is called sequential sort or selection sort algorithm and the other is called the ***Bubble Sort Algorithm*** or exchange sort algorithm. The worst case complexity of this algorithm is ***O(n1ogn).***

NC Algorithms

There has been a considerable interest in problems which have parallel algorithms running in time bound by a polynomial in the logarithms of the size of the input and using a number of processors polynomial in the input size.

Such a class of problems referred to as the class ***Non Deterministic Polynomial Complete*** (NC). There is a greater amount of interest in the NC because the speed of the NC algorithms is exponential compared to their sequential versions. Furthermore, they make use of a reasonable amount of hardware, namely, the processo1,s. The class NC is 'robust' under reasonable changes in the underlying machine models of parallel computation. Recently, NC algorithms have been gken for recognizing comparability graphs, permutation graphs, and interval graphs algorithm for recognizing chordal graphs and k-trees are of greater importance in the research area. These algorithms use 0(n4) processors and take O(1ogn) time on the PRAM model of computation.

NP - Completeness

Class P is defined to be the set of problems which can be solved by algorithms of polynomial time complexity. Class NP is defined to be the set of problems for which possible solutions can be verified in polynomial time. Algorithms for solving problem:; in this class are known as 'Nondeterministic' algorithms, and it consists of two stages: the 'guessing' stage in which the possible solutions are made available, and the 'checking' stage in which it is determined whether each suck possibility is in fact a solution. For NP problems the checking stage is required to be polynomially bounded. The term 'NP' represents the capacity to solve problems in polynomial time on a nondeterministic Turing Machine, as discussed by Garay & Johnson .

Every problem in class I' is also in NP, that is, P ***c*** NP. This is known by observation that any of the algorithms in P can be used as the checking stage of some nondeterministic algorithms**.** It is not known whether there are problems in NP which are not in P. The issue of whether P = NP remains an open question, although it is widely believed that P *c* NP. A large number of problems in NP, (there are well over 300 listed ,have been proved equivalent in terms of complexity in the sense that if one of these problems is in P then they are all in P. These problems are said to be NP - ***Complete.*** This. equivalence is based on polynomial time reducibility, whose problem L is reducible to problem M if every instance of problem L can be uniquely transformed by an algorithm of polynomial complexity into an instance of M. **A** more precise definition of class NP -complete is that it is the set of problems L such that L is in NP, and all other problems in NP can be polynomial time reduced to L.

CONCLUSION:

Polysemic intersection representation for Bipartite Graphs, Subtree graphs, and Path graphs have been defined and linear time algorithms for the recognition of polysemic intersection pairs of Interval graphs, Circular – arc graphs, Vertex path graphs, and Perfect vertex path graphs have been developed. The algorithm mainly rely on the graph theoretical model, transitive tournament. When we consider **;L** tree as the underlying structure and the family of sets correspond to paths in the tree, the tree can be thought of as a computer communication network with paths representing messages between users. Also the edge path has transmission links as the

bottleneck in the computer network example, and the vertex path has switching nodes as the bottleneck.

Finding a Hamiltonian circuit in the Joint graph of two intersection graphs in the same vertex set, is a1r.o found as the characteristics to recognize a polysemic intersection pair. The clique tree representation property for the polysemic intersection pairs has, been discussed. Clique tree representation is used in various computer algorithms. The strongly chordal property of polysemic intersection pair of PV graphs is also introduced. Graph polysemy is of great importance in both theoretical and practical aspects. There are a number of open problems in the field of polysemic representation of graphs and the recognition of polysemic intersection pairs of various graphs. The polysemic inte-section number is the minimum cardinality of the universal set in polysemic ir~tersection representation. The problem of finding this number is also open.

Interesting directions for further stutiy are given below.

(1) Algorithms for recognizing palysemic pair of other classes of graphs.

(2) Construct polysemic intersec ion representation in 0(n4) time.

(3) Determine for an arbitrary pair of intersection polysemic graphs, the exact value of their polysemi: intersection number.

(4) Find combinatorial techniques to determine for an arbitrary pair of polysemic intersection graphs, the exact value of their polysemic intersection number.

(5) Find a polysemic intersectioq representation with respect to some set of size 0 (n3).

REFERENCES:

- A.V. Aho, J. E. I3opcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, Reading, Massachusetts.
- A.V. Aho, J. E. -Iopcroft, and J. D. Ullman, Data structures and Algorithms, Addison Wesley, Reading.
- A.V. Aho, J.E. Hopcroft, and J.D.Ullman, Data structures and Algorithms, Addison Wesley, Reading.
- Ian Anderson, A first Course in Combinatorial Mathematics, Clarendon Press, Oxford.
- Dr. A. Antcnysamy S.J., An algorithmic study of some classes of Intersection Graphs, Ph.D. Thesis, Kamaraj University, Madurai .
- Bertossi and Maurizio A. Bonuccelli, Hamiltonian circuits in Inter~agl raph generalizations, Information Processing, Lett:rs,23, (1986), 195 - 200.
- Bertossi and Maurizio A. Bonuccelli, Some Parallel Algorithms on Interval graphs, Disc. App. Math.
- Bondy and 'I. Chvatal, A method in graph theory, Discrete Math. 15 (1976) 11 1-136.
- Berge and V. Chvatal (Eds.), Topics on Perfect graphs, Ann. Of Discrete Math.
- Berge, Some classes of perfect graphs, in Graph Theory and Theoretical Physics, Academic Press, 155-165.
- Berge, Theory of graphs, North - Holland, Amsterdam.