ALGORITHMS FOR SOME INTERSECTION GRAPHS

T. Kashiwabara
Dept. of Information and Computer Sciences
Faculty of Engineering Science, Osaka University
Toyonaka, Japan 560

Abstract. Several intersection graphs such as curves-in-the-plane graphs, circular-arc graphs, chordal graphs and interval graphs are reviewed, especially on their recognition algorithms. In this connection graph realization problem is mentioned.

1. Introduction

This report summarizes some already known results on certain intersection graphs. A finite undirected graph G=(V,E) is an intersection graph on intersection model $M=(S,\mathcal{S})$ when there is a 1-1 correspondence f between V and $\mathcal{S}$ such that $(u,v) \in E$ iff $f(u) \cap f(v) \neq \phi$, where $\mathcal{S}$ is a collection of subsets of S. Trivially, any graph is an intersection graph on some appropriate model M [1]. In practical applications, graphs having simple intersection models such as chordal graphs and interval graphs etc. are of interest [2][3].

In the following, several intersection graphs having particular models are considered. They are curves-in-the-plane, chordal, path, interval, circular-arc and circle graphs. Relations between them are mentioned and recognition algorithms and computational complexity of several problems on them are sketched. Connection between path graphs and the graph realization problem is also mentioned.

2. Curves-in-the-plane graphs and circle graphs

Without loss of generality, the set S in intersection model $M=(S,\mathcal{S})$ can be taken to be a plane: Any graph is an intersection graph on the plane. Therefore it is necessary to restrict the family of subsets to have an appropriate property for the purpose of defining a proper subclass of graphs. When every $S' \in \mathcal{S}$ is a connected piece in the plane, the corresponding intersection graphs seem to be a proper subclass of all graphs. In [4], the class of intersection graphs of curves in the plane is introduced.

Definition. Let V be a set of (non self-intersecting) curves in the plane. The corresponding curves-in-the-plane graph has V as the vertex set and two vertices are connected by an edge iff two corresponding curves intersect in the plane. Similarly straight-lines-in-the-plane graph is defined as an intersection graph of straight line segments in the plane.

The following facts are reported.

Theorem 1 General graphs $\supsetneq$ curves-in-the-plane graphs $\supsetneq$ straight-lines-in-the-plane graphs [4].

Fig.1 shows an example which is not a curves-in-the-plane graph, as underlying K_5 is not planar.

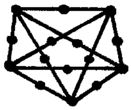


Fig.1

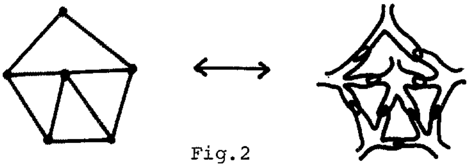


Fig.2

Theorem 2 Every planar graph is a curves-in-the-plane graph [4]. (see Fig. 2)

Theorem 3 The problem of finding the chromatic number for straight-lines-in-the-plane graph is NP-complete [4].

The class of circle graphs is a restricted class of straight-lines-in-the-plane graphs.

Definition A circle graph is an intersection graph of straight line segments in the circle such that two endpoints of each line segment are on the circle.

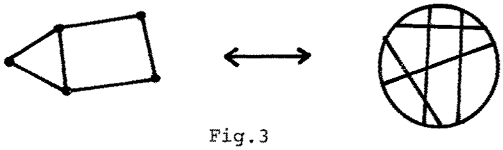


Fig.3

Complexity of coloring a circle graph is reported.

Theorem 4 Determining the chromatic number of circle graphs is NP-complete [6].

No recognition algorithms for curves-in-the-plane graphs, straight-lines-in-the-plane graphs and circle graphs have been given so far.

3. Circular-arc graphs

Definition A graph is a circular-arc graph iff it is the intersection graph of arcs on a circle.

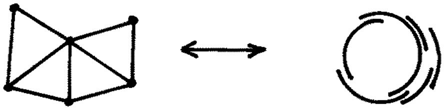


Fig.4

The class of circular-arc graphs is a proper superset of interval graphs. Though there exists a polynomial time algorithm for determining the chromatic number of interval graphs, the same problem for circular-arc graphs seems intractable. There is reported the following.

Theorem 5 Determining the chromatic number for circular-arc graphs is NP-complete [6].

Unlike chordal graphs, the number of dominant cliques of a circular-arc graph or a circle graph (which is a curves-in-the-plane graph) can grow exponentially. Fig.5 shows an example of such a graph [7][8]. It is a complete n-partite graph of 2n vertices which has 2^n dominant cliques.

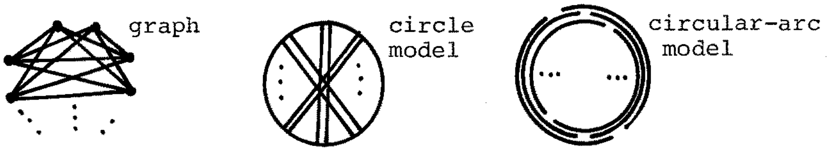


Fig.5

Because of the above fact, efficient recognition algorithms for circle or circular-arc graphs, if exist, cannot use the set of all dominant cliques. This is essentially different from the recognition algorithm for interval graphs which makes use of dominant cliques effectively. An efficient recognition algorithm for circular-arc graphs which searches for an appropriate order of vertices is presented in [8].

4. Subtree graphs (chordal graphs)

In sections 4-6, intersection model on a tree is treated, i.e., the case where the set S in model M is a tree. Also for this case, an arbitrary undirected graph can be an intersection graph if disconnected pieces for $S' \in \mathcal{S}$ are allowed. Restricting the sets in $\mathcal{S}$ to be connected leads to the following definition.

Definition A graph is a subtree graph iff it is the intersection graph of subtrees in a tree. (a tree or a subtree is thought to be connected.)

Related to the Gaussian elimination pivot ordering of symmetric matrices, chordal graphs are known.

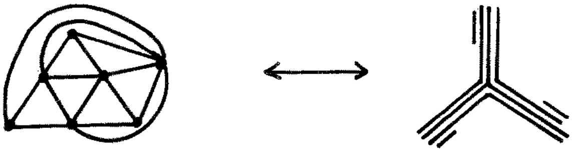


Fig.6

<u>Definition</u> A graph is a chordal graph iff every cycle with more than 3 edges has a chord.

In [9], it is proved that the above two classes of graphs coincide with each other.

<u>Theorem 6</u> A graph is a subtree graph iff it is a chordal graph [9].

Any subtree graph has a, so called, rooted tree model as follows.

<u>Definition</u> A model $M=(S,\mathcal{S})$ for a subtree graph is a rooted tree model if

i) the underlying tree S has a root on the left and stretches to the right, i.e., any branch point of the tree S has only one edge incident from the left

ii) the function $x(v)$ is one to one where $x(v)$ is defined as follows. Let $l(v)$ be the point on the tree S which is the leftmost point of the subtree corresponding to vertex v. ($l(v)$ is well defined because of the property of tree S). $x(v)$ is defined as the x-coordinate of $l(v)$.

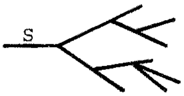


Fig.7

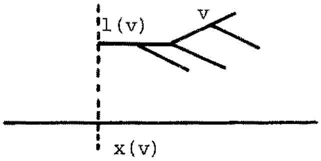


Fig.8

Rooted tree models are not restrictive.

<u>Lemma</u> Let G be any subtree graph and v_1 be any vertex of G. G has a rooted tree model for which $x(v_1)$ is the smallest value among $\{x(v)\}_{v \in V}$.

There is a recognition algorithm for chordal graphs which is based on LBFS [10]. It can be interpreted on the above mentioned intersection models as follows. First, a chordal order can be defined on a rooted tree model.

<u>Definition</u> $v_n, v_{n-1}, \ldots, v_1$ is a chordal order if, for any i ($1 \leq i \leq n$), v_i is a simplicial vertex for the section graph $G(\{v_{i-1},\ldots,v_1\})$, that is, any distinct vertices u,v in $\{w \in V \ ; \ w \in Adj(v_i) \cap \{v_{i-1},\ldots,v_1\}\}$ are connected by an edge.

<u>Theorem 7</u> $v_n, v_{n-1},\ldots,v_1$ is a chordal order of vertices iff $x(v_n) >$

$x(v_{n-1}) > \ldots > x(v_1)$ for some rooted tree model.
This is because subtrees corresponding to u and v must pass the point $l(v_i)$ in the corresponding rooted tree model. (see Fig.9)

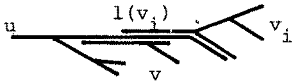


Fig.9

A recognition algorithm for subtree graphs can be divised which tries to construct rooted tree models from the left to the right, in particular, left to right order of $l(v)$ points. Any vertex, say v_1, can be put at the leftmost position on the tree S. The detail of the form of the subtree v_1 and tree S is not determined at this stage. Only the point $l(v_1)$ is fixed. Consider three vertices u,v,w such that u,v, $w \in Adj(v_1)$, $(u,v) \notin E$, $(u,w) \in E$ and $(u,w) \in E$.

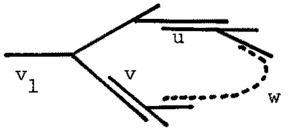


Fig.10

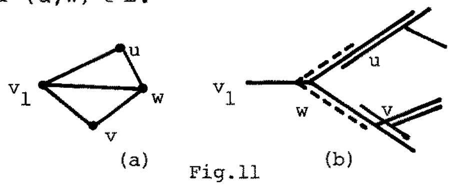


Fig.11

If $(w,v_1) \notin E$, then there exists no intersection model for such a graph and the construction of a model stops. Now assume $(w,v_1) \in E$. Suppose w is to be located on the tree after putting u and v. Then x(w) must be smaller than x(u) or x(v). Thus putting v_1,u,v,w in such a sequence destroyes left-to-right-rule. This suggests putting vertices in lexicographic breadth first order. Actually, the following theorem is given which is an intersection model interpretation of LBFS recognition algorithm for chordal graphs [10].

Theorem 8 A rooted tree model for a subtree graph can be constructed by any LBFS sequencing.

LBFS on a graph can be performed in linear time, so the recognition algorithm for chordal graphs [10].

For subtree graphs, the number of dominant cliques is not greater than the number of vertices. This can be used in recognition algorithms for interval graphs and path graphs in the following sections, though it is immaterial for subtree graph recognition.

Unlike the case for circular-arc graphs, the determination of chromatic number of a chordal graph can be performed in polynomial time [11].

5. Path graphs and graph realization problem

Definition A path graph is an intersection graph of simple paths in a tree.

The class of path graphs is a subclass of subtree graphs, and is a superclass of interval graphs.

The above inclusion is strict as is seen from Fig. 12.

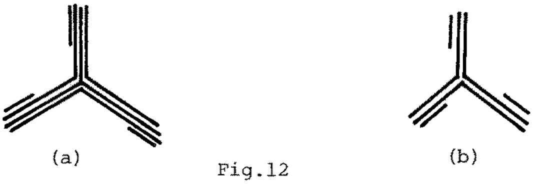


Fig.12

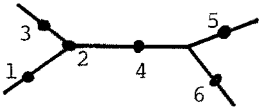
No fast recognition algorithm for path graphs seems to be reported.

A path graph is a subtree graph and so the set of dominant cliques can be obtained in linear time. As in the case for interval graphs, recognition for path graphs can be reduced to a placement problem of dominant cliques.

Here a placement problem on a tree is defined.

Definition (Linear placement on a tree)

Given a set of n points $S=\{s_1,...,s_n\}$ and a family $\mathcal{S}$ of subsets p_i of S $(i=1,...,k)$, place $s_1,...,s_n$ on an appropriate tree so that elements of every p_i appear consecutively on a path on the tree.



$S=\{1,2,3,4,5,6\}$
$p_1=\{1,2,4,5\}$
$p_2=\{1,2,3\}$
$p_3=\{2,3,4,6\}$
$p_4=\{5,6\}$

Fig.13

Now the recognition for path graphs can be stated using the above definition.

Theorem 9 A subtree graph G is a path graph iff there is a solution for linear placement on a tree problem for which S is the set of dominant cliques and p_i is the set of dominant cliques including vertex v_i, such that full branch point constraint is satisfied. Here full branch point constraint requires that every branch point of the tree has a point $s_j \in \mathcal{S}$.

The last constraint is needed because each S_i represents a dominant clique and $\{S_i\}$ is the set of all dominant cliques.

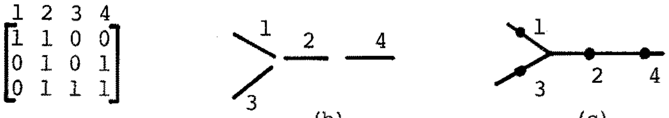
There is a similar placement problem in graph realization problem.

Definition (Graph realization problem)

Given a 0-1 matrix K, find a graph which has K as the principal part of the fundamental circuit matrix. In a fundamental circuit matrix [I:K], each row corresponds to a circuit, and each column to an edge.

This problem is equivalent to the next placement problem: Place columns of K on a tree so that for every row r, R(r) (the set of columns whose r-th row is 1) appears consecutively on a simple path on the tree. Every branch point must not have a column.

The last constraint is due to the fact that every column represents an edge. (see Fig.14)



$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ \hline 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{array}$$

(a) (b) (c)

Fig.14

So to say, graph realization problem is a linear placement on a tree problem under empty branch point constraint.

Thus, path graph recognition and graph realization problem are different only in the constraints on the branch points.

Though there seems no simple reduction between these two problems, the algorithm for graph realization problem in [15][16] seems to be efficiently used for path graph recognition.

In section 4, it is shown that a graph is a subtree graph iff it has no chordless cycle (with more than 3 edges). This is a kind of forbidden graph formulation of graphs. Such a formulation will be given for interval graphs in the following section.

Graph realization problem is known to have such a kind of formulation. However, it seems that no forbidden graph formulation has been given for path graphs.

6. Interval graphs

An interval graph is a subtree graph which has a intersection model whose underlying tree is a line. (i.e. without branch point)

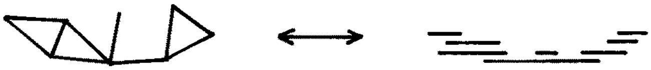


Fig.15

As in subtree graphs, l(v) and x(v) of the model define an ordering of vertices [12]. Using such an ordering, a recognition algorithm for interval graphs can be made [12][13], though no linear time algorithm has been obtained in this direction yet.

Linear time test for interval graphs now available runs in two steps [14]:

1 Perform chordal graph test and obtain a vertex vs. dominant clique matrix.

2 Test the matrix obtained above for consecutive ones property using PQ tree algorithms.

Step 1 can be performed using LBFS and step 2 can be performed in linear time using a sophisticated data structure called PQ trees.

Next is a structure theorem for interval graphs [22].

Definition Three vertices a_1, a_2, a_3, of a graph is called an asteroidal triple if there exist three paths w_1, w_2, w_3 such that for i=1,2,3, (i) w_i connects the two vertices a_j $(j \neq i)$ (ii) a_i is not adjacent to any vertices in w_i.

Theorem 10 A graph is an interval graph iff it is chordal and has no asteroidal triple.

In [22], interval graphs are characterized by five kinds of forbidden graphs. No linear time recognition algorithm is given using such a structure yet.

7. Completion problems and isomorphism

Completion problem is to obtain a minimum number of edges to be added so that the augmented graph becomes to have a specified property. Completion problems are studied for several intersection graphs mentioned before.

Theorem 11 Completion problems for interval graphs and path graphs are NP-complete [6][19][20].

Whether other completion problems (for chordal graphs, for circular-arc graphs and for curves-in-the-plane graphs) are NP-complete remains unsolved.

For isomorphism test, the next results are known.

Theorem 12 Isomorphism test for general graphs, isomorphism test for chordal graphs and isomorphism test for path graphs are polynomially reducible to one another [6] [21].

Theorem 13 There is a polynomial time isomorphism test for interval graphs [21].

8. Relations among the classes

Several relations among the various classes are known.

Theorem 14

$$\begin{tikzcd}
\text{graphs} \arrow[r, phantom, "\supsetneqq"] & \text{curves-in-the-plane} \arrow[r, phantom, "\supsetneqq"] \arrow[dd, phantom, "\rotatebox{-90}{$\supsetneqq$}"] & \text{straight-lines-in-the-plane} \arrow[r, phantom, "\supsetneqq"] \arrow[d, phantom, "\rotatebox{-90}{$\supsetneqq$}"] & \text{circle} \\
 & & \text{circular-arc} \arrow[dr, phantom, "\rotatebox{-60}{$\supsetneqq$}"] & \\
 & \text{subtree(chordal)} \arrow[r, phantom, "\supsetneqq"] & \text{path} \arrow[r, phantom, "\supsetneqq"] & \text{Interval}
\end{tikzcd}$$

Some explanation would be necessary. In the definition of curves-in-the-plane graphs, each curve has no self-intersecting point and two curves have only finite number of intersecting points. This seems different from the definition of other classes, for example a circular-arc model may have a circle as a subset and the number of intersecting points may be infinite.

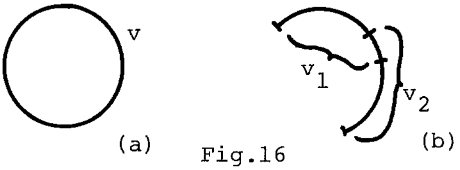


Fig.16

However these differences are immaterial as is seen below. For a circle in circular-arc model, cutting at some appropriate point does not change the intersection graph. Then draw the model on the plane as in fig.17 (b) so that each arc does not intersect the others. (this is the usual illustration for circular-arc model, though the meaning is quite different). Now modify each circle to obtain necessary intersection by zigzagging sufficiently.

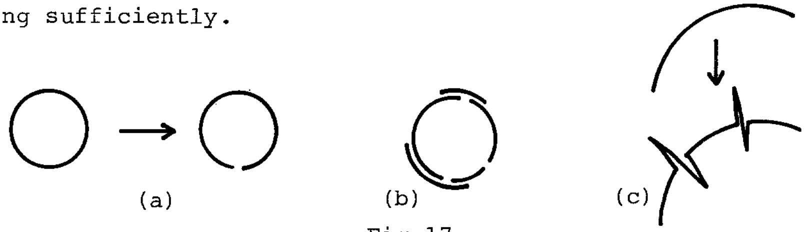


Fig.17

The same configuration applies to subtree graphs. In this case each subtree is transformed once to a fringing curve as in fig.18(b) (just as in [4]) and then modified to have zigzags to intersect necessary curves.

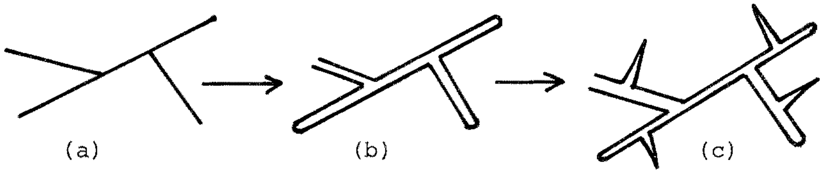


Fig.18

9. Conclusion

Fast recognition algorithms are known for several intersection graphs. However many completion problems are still open.

The author wishes to thank Professor T. Fujisawa for his helpful suggestion and encouragement.

References

[1] F. Harary, Graph theory, Addison-Wesley, Reading Massachusetts (1972).

[2] D.J. Rose, Triangulated graphs and the elimination process, J. Math. Anal. Appl., 32, 597-609 (1970).

[3] T. Ohtsuki, H. Mori, E.S. Kuh, T. Kashiwabara and T. Fujisawa, One-dimensional gate assignment and interval graphs, IEEE tran. on CAS Vol. CAS-26, No.9, 675-684 (1979).

[4] G. Ehrlich, S. Even and R.E. Tarjan, Intersection graphs of curves in the plane, J. Combinatorial theory (B), 21, 8-20 (1976).

[5] S. Even and A. Itai, Queues, stacks and graphs, Theory of machines and computations, Z. Kohavi and A. Paz, ed., Academic Press, New York, 71-86 (1971).

[6] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, H. Freeman and Sons, San Francisco (1978).

[7] F. Gavril, Algorithms for a maximum clique and a maximum independent set of a circle graph, Network, 3, 261-273 (1973).

[8] A. Tucker, An efficient test for circular-arc graphs, SIAM J. Comput., Vol.9, No.1, 1-24 (1980) .

[9] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, J. Combinatorial theory (B), 16, 47-56 (1974).

[10] D.J. Rose, R.E. Tarjan and G.S. Lueker, Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput., Vol.5, No.2, 266-283 (1976).

[11] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum covering by cliques and maximum independent set of a chordal graph, SIAM J. Comput., 1, 180-187 (1972).

[12] T. Oyamada and T. Ohtsuki, Interval graphs and layout design of MOS arrays, IECE technical report CST75-83 (1975) (in Japanese).

[13] T. Yoshida, An algorithm for obtaining a perfect order of an interval graph, monograph, Osaka University (1976) (in Japanese).

[14] K.S. Booth and G.S. Lueker, Testing for the consecutive ones property,

interval graphs and graph planarity using PQ-tree algorithms J.Computer and System Sciences, 13, 335-379 (1976).

[15] S. Fujishige, An efficient algorithm for solving the graph-realization problem by means of PQ-trees, Proceedings of 1979 ISCAS, IEEE catalog No.79 CH 1421-7 CAS. 1012-1015 (1979).

[16] S. Fujishige, An efficient algorithm for solving the graph-realization problem, IECE technical report CST78-136 (1979) (in Japanese).

[17] A. Lempel, S. Even and I. Cederbaum, An algorithm for planarity testing of graphs, in "Theory of graphs: international Symposium: Rome, July, 1966" (P. Rosenstiehl, Ed.), 215-232, Gordon and Breach, New York (1967).

[18] K.S. Booth, PQ-tree algorithms, Ph.D. Dissertation, Department of Electrical Engineering and Computer Sciences, University of California Berkeley, California (1975).

[19] T. Kashiwabara and T. Fujisawa, An NP-complete problem on interval graph, Proceedings of 1979 ISCAS, 82-83 (1979).

[20] T. Kashiwabara, Y. Masaki and T. Fujisawa, Complexity of intervalization of graphs, IECE technical report CST77-15 (1977) (in Japanese).

[21] G.S. Lueker, Efficient algorithms for chordal graphs and interval graphs, Ph.D. Dissertation, Program in Applied Mathematics and the Department of Electrical Engineering, Princeton University, Princeton N.J., (1975).

[22] C.G. Lekkerkerker and J.Ch. Boland, Representation of a finite graph by a set of intervals on the real line, Fundamenta Mathematicae, 51, 45-64 (1962).