Oliver Tonnesen
V00885732

Project 4

1 Introduction

1.1 Background

An interval graph is a graph related to a set of intervals on the real line. The interval graph problem was first posed by György Hajós in 1957. [**?**] The problem is to determine whether or not a given graph is isomorphic to an interval graph.

1.2 Applications

Interval graphs can be applied very naturally to many types of scheduling and resource allocation problems:

If each interval represents a task to be completed, and certain tasks (namely those with intersecting intervals) cannot be done at the same time, then finding an independent set in the corresponding interval graph is equivalent to finding a set of tasks that can be completed simultaneously.

If each interval represents a course needing a classroom at a particular time, then finding an optimal colouring of the corresponding interval graph represents an allocation of the fewest possible classrooms such that no two courses use the same classroom at the same time.

When compiling a computer program to machine code, the compiler must store a number of variables' values in a small number of registers. If two variables are never used simultaneously, then they can be allocated to the same register. The compiler can check when the first and last use of each variable occurs, and use these intervals to create an interval graph. The minimum number of registers required to run the program is equal to the chromatic number of the this interval graph.

1.3 Definitions

Interval Graph: Given a family of intervals $\{S_i\}_{i=1,\ldots,n}$, the corresponding interval graph is defined by assigning a vertex v_i to each S_i, and joining v_i and v_j with an edge whenever $S_i \cap S_j \neq \emptyset$.

Comparability Graph: A comparability graph is an undirected graph connecting pairs of elements that are comparable in a partial order.

Clique Matrix: Given a graph G, its clique matrix M has rows representing maximal cliques in G, and columns representing vertices in $V(G)$, where $M_{ij} = 1$ if vertex j is in clique i, and 0 otherwise.

2   Characterizations

2.1   Vertex ordering

A graph with maximal cliques $M_1, \ldots, M_k$ is an interval graph if and only if its cliques can be ordered in such a way that for any vertex v, the maximal cliques containing v occur consecutively. [**?**]

2.2   Forbidden subgraphs/structures

A graph is an interval graph if and only if every induced cycle has exactly three vertices, and its complement is a comparability graph. [**?**]

2.3   Clique covering

A graph is an interval graph if and only if the rows of its clique matrix can be permuted such that the 1s in each column appear consecutively. [**?**]

3   Recognition algorithms

Booth and Lueker [**?**] showed in 1976 that given a finite set X and a collection $\mathcal{F}$ of subsets of X, the question of whether or not there exists a permutation of X in which the elements of each subset in $\mathcal{F}$ appear consecutively as a subsequence of the permutation can be decided in linear time. This can be applied to the interval graph problem as follows:

Given a graph G, let X be the set of maximal cliques of G, and let $\mathcal{F} = \{C_v \mid v \in V(G)\}$, where C_v is a maximal clique containing v. If there does exist a permutation of X as described above, then by the first characterization of interval graphs we gave, G is an interval graph, and similarly if no such permutation exists, then G is not an interval graph.

4   Optimization problems

4.1   Colouring

It's clear to see that the minimum number of colours needed to properly colour an interval graph is the same as the largest set of pairwise intersecting intervals. In fact, if we order the vertices of an interval graph according to the left endpoint of their interval, then a greedy colour will colour the graph optimally. Importantly, this procedure allows an interval graph to

be coloured in linear time, whereas this problem is extremely difficult in a general graph. [**?**]

4.2   Maximum clique

Every maximum clique in an interval graph is a maximum set of pairwise intersecting intervals. As we mentioned above, the largest set of pairwise intersecting intervals has the same size as the number of colours required to properly colour the interval graph. So the obvious way to find a maximum clique in an interval graph is to greedily colour it using the above given ordering, and to take a subsequence $v_1, \ldots, v_\chi$ of the ordering such that v_1 is coloured 1, and v_{i+1} is the first vertex after v_i to be coloured $i + 1$. v_{i+1} must be adjacent to all previous vertices in the subsequence, as if it weren't adjacent to one of them, say v_k, $k < i + 1$, the greedy algorithm would have coloured v_{i+1} with the colour k instead. So our sequence does indeed form a clique, and that clique has size χ.

4.3   Maximum independent set

Our second characterization of interval graphs mentioned that any induced cycle has exactly three vertices. Graphs of this kind are also called chordal graphs. In 1972, Gavril presented a linear time algorithm for finding a maximum independent set in a chordal graph. [**?**]

4.4   Minimum clique covering

A minimum clique covering of any graph must have at least as many vertices as a maximum independent set. Since we're considering an interval graph, our maximum independent set must intersect all other vertices. The existence of a vertex not in the independent set and not intersecting the intervals of any vertices in our independent set would violate the maximality of our independent set. Similarly, the existence of two nonintersecting intervals both lying inside the interval of a vertex in our independent set would also violate the maximality of our independent set. So each vertex uniquely covers one of each of the maximal cliques in our graph, and thus determines a clique covering. So any minimum clique covering must have at most as many cliques as there are vertices in our independent set. Thus in an interval graph, the minimum clique covering contains exactly as many cliques as the number of vertices in a maximum independent set.