

$\chi(\langle Y \rangle) \geq 3$. By (1),
action, if $z_i \in Z$, then
 $\chi(\langle Y \cup Z \rangle) \geq 3$.
orem.

Weighted Graphs," *Proc.*
Information Sciences

of a Graph," *Proc.*
pp. 1142-1146.

Graphs of Invulnerability
on Circuit Theory,

Invulnerability of the
SIAM J. App. Math.

Construction of Least Cost
Graph with the Maximum
Circuit Theory, Vol.

with Given Connections
with Given Measures of
E Transactions on Circuits
2-10.

ley, Reading, Mass.

orce Office of Defense
AF-AFOSR-71-2151,

Algorithms for a Maximum Clique and a Maximum Independent Set of a Circle Graph

Received
The Weizmann Institute of Science
Rehovot, Israel

ABSTRACT

Consider a family of chords in a circle. A circle graph is defined by representing each chord by a vertex, two vertices being connected by an edge when the corresponding chords intersect. In this paper, we describe efficient algorithms for finding a maximum clique and a maximum independent set of circle graphs. These algorithms require at most n^3 steps, where n is the number of vertices in the graph.

Keywords: Circle graph, transitive graph, interval graph, permutation graph, overlap graph, maximum clique, maximum independent set.

1. INTRODUCTION

In this paper, we consider only finite graphs with no parallel edges and no self-loops. The *complement* of a graph is a graph with the same set of vertices as the given one, two vertices of the complement being connected if and only if they are not connected in the given graph. A *subgraph* of a graph, determined by a subset of the graph's set of vertices, two vertices of the subgraph being connected by an edge, if and only if the two vertices are connected in the given graph. A set of vertices in a graph is called *independent* if no two of its elements are connected. A *maximum independent set* is one with the largest number of vertices of all independent sets. A *clique* is a maximal, completely connected set of vertices; a *maximum clique* is one of a maximum size.

If two vertices u, v are connected by an undirected edge, we denote it by $u-v$, and if they are connected by a directed edge

from u to v , we denote it by $u \rightarrow v$. The set of vertices connected to a vertex v is denoted by Γv . In a directed graph, we denote:

$$\Gamma v^{-1} = \{u \mid u \rightarrow v\}.$$

For a set A , $|A|$ will denote the number of elements in A .

Consider a system of chords in a circle, and draw a graph in the following way: every vertex represents a chord, and two vertices are connected by an edge when the corresponding chords intersect. A graph which can represent a system of chords in the above manner is called a *circle graph*. For example, the graph in Figure 1b represents the family of chords of the circle in Figure 1a. The circle graphs were introduced by Even and Itai in [1]. These graphs are obtained from permutations, and the authors showed that the chromatic number of a circle graph is equal to the minimum number of parallel stacks realizing the corresponding permutation. We can see the family of chords also as a system of roads, and then, look for the maximum number of roads no two of which intersect. We obtain this by finding a maximum independent set in the corresponding circle graph.

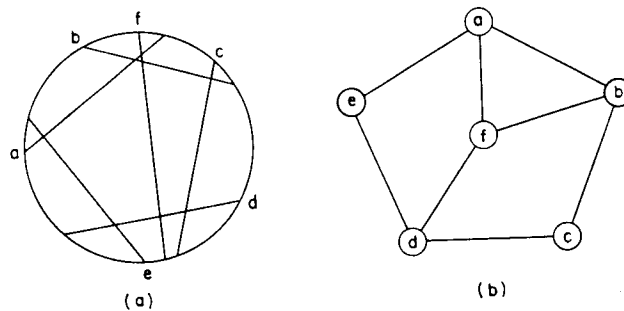


Fig. 1

The purpose of this paper is to give efficient algorithms for finding a maximum clique and a maximum independent set of a circle graph. These problems are not trivial, since there are families of circle graphs whose number of cliques or independent sets grow exponentially with the number of vertices.

In Figure 2a there are $\frac{n}{3}$ triples of chords $\{a_i, b_i, c_i\}_{i=1}^{\frac{n}{3}}$. For $i \neq j$, a_i, b_i and c_i intersect a_j, b_j and c_j ; but each two of a_i, b_i, c_i do not intersect. Every clique of the circle graph

corresponds to a sequence of chords which contains exactly one chord from every triple, thus, there are $3^{\frac{n}{3}}$ cliques. In a similar way, the graph of the family in Figure 2b has $3^{\frac{n}{3}}$ independent sets.

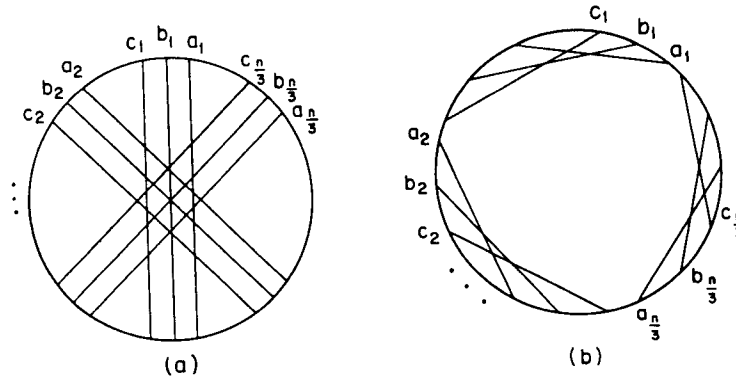


Fig. 2

Consider two chords with a common endpoint in a given family of chords. We can slightly move one chord without changing the intersecting relation and the representing graph, so that the two chords do not have a common endpoint anymore. Therefore, we can assume without loss of generality that no family of chords, we deal with, contains chords with common endpoints.

First of all, let us describe some known algorithms that we will need. A directed graph is *transitive* if it has no directed circuits, and for every three vertices u, v, w the existence of the edges $u \rightarrow v$ and $v \rightarrow w$ implies $u \rightarrow w$. Let G be a transitive graph with n vertices. We can rename the vertices by $1, 2, \dots, n$ so that every directed edge will be directed from a vertex named with a low number to a vertex named with a high number. This can be done in the following way. Since G is finite and has no directed circuits, it must have a sink (a vertex with no outgoing edges). We take a sink v , and rename it n . Now, we delete it with its adjacent edges from G ; let G_1 be the remaining subgraph. G_1 is also circuit free and we rename one of its sinks by $n-1$. By deleting this vertex we obtain G_2 , and so on. Finally, G_n is empty. In this way we renamed all the vertices of G so that the edges go from low to high.

vertices connected graph, we denote:

ents in A. and draw a graph a chord, and two corresponding chords n of chords in example, the rds of the circle d by Even and mutations, and a circle graph ks realizing the ly of chords he maximum number: this by finding circle graph.

cient algorithms dependent set of al, since there cliques or inde- er of vertices.

$\{a_i, b_i, c_i\}_{i=1}^{\frac{n}{3}}$. For each two of he circle graph



In [2], an efficient algorithm is given for finding a maximum clique of a transitive graph. It works as follows. Consider the transitive directed graph G whose vertices were renamed by $1, 2, \dots, n$ so that the edges go from low to high. Let us denote the subgraph defined on the set of vertices $\{1, 2, \dots, i\}$, $1 \leq i \leq n$ by G^i . To every vertex i , we will attach a set of vertices S_i , which is a maximum clique containing i in G^i , and a number $c(i) = |S_i|$. Clearly, $S_1 = \{1\}$ and $c(1) = 1$. Assume by induction that S_1, S_2, \dots, S_{i-1} and $c(1), c(2), \dots, c(i-1)$ are known. If $\Gamma_i^{-1} = \phi$, then $S_i = \{i\}$ and $c(i) = 1$. If $\Gamma_i^{-1} \neq \phi$, let $\Gamma_i^{-1} = \{i_1, \dots, i_k\}$; clearly $i_1, i_2, \dots, i_k < i$. For every $i_j \in \Gamma_i^{-1}$, $1 \leq j \leq k$, $S_{i_j} \cup \{i\}$ is a clique of G^i . For if $m \in S_{i_j}$, $m \neq i_j$ then $m \rightarrow i_j$ and $i_j \rightarrow i$; thus, by the transitivity of G^i , $m \rightarrow i$. On the other hand, every clique containing i of G^i is obtained by annexing i to a clique of G^{i_j} . Let $i_r \in \Gamma_i^{-1}$ be a vertex for which $c(i_r) = \max_{j \in \Gamma_i^{-1}} c(j)$; hence $|S_{i_r}| = \max_{j \in \Gamma_i^{-1}} |S_j|$. Then $S_{i_r} \cup \{i\}$ is a maximum clique containing i in G^i , and we define $c(i) = c(i_r) + 1$, $S_i = S_{i_r} \cup \{i\}$. Assume we constructed the sequences $c(1), \dots, c(n)$ and S_1, \dots, S_n , and let k be a vertex for which $c(k) = \max_{1 \leq i \leq n} c(i)$. Then S_k is a maximum clique of G and $c(k)$ is the number of its vertices.

This algorithm can be extended to a weighted graph, where a weight $\omega(i)$ is assigned to every vertex i , and we look for a maximum weight clique (a clique with the maximum sum of the weight of its vertices). In this case we construct the sequences $c(1), \dots, c(n)$ and S_1, \dots, S_n , by:

$c(1) = \omega(1), S_1 = \{1\};$

if $\Gamma_i^{-1} = \phi$ then $c(i) = \omega(i)$ and $S_i = \{i\};$

if $\Gamma_i^{-1} \neq \emptyset$ and for a vertex k , $c(k) = \max_{j \in \Gamma_i^{-1}} c(j)$,

then $S_i = S_k \cup \{i\}$ and $c(i) = \omega(i) + c(k)$.

The above algorithms take no more than n^2 steps.

Let a line be drawn from the left to the right. Consider a family F of intervals on this line and construct a graph G in the following way: every vertex v of G represent an interval $\bar{v} \in F$, and two vertices are joined by an edge when the two corresponding intervals intersect. Such a graph is called an *interval graph* (see [4], [5] and [6]). Let the graph G' denote the complement of G . For any edge $u-v$ in G' , the intervals \bar{u}, \bar{v} are disjoint; then, we direct $u \rightarrow v$ in G' if and only if \bar{u} appears to the left of \bar{v} on the line. This is a transitive orientation of G' , since it is circuit free, and also: if $u \rightarrow v$ and $v \rightarrow w$ then \bar{u} is disjoint and is placed on the left of \bar{w} , \bar{u} is disjoint and is placed on the left of \bar{v} , thus \bar{u} is disjoint and is placed on the left of \bar{w} , implying $u \rightarrow w$. Therefore, the complement of an interval graph is transitively orientable. Clearly, a set of vertices is an independent set for G if and only if this set is a clique for the complement graph G' . Therefore, for finding a maximum independent set of G we apply the above algorithm for finding a maximum clique of G' .

A graph is *chordal*, if every simple circuit with more than three vertices has an edge connecting two non-consecutive vertices. These graphs are discussed in [3] and [7]. It is known (see [4] and [6]), that every interval graph is chordal. We must remark that the circle graph of Figure 1b is not chordal and not transitively orientable, since it contains the pentagon $\{a,b,c,d,e\}$ as a subgraph. Thus, the algorithms of [2] and [7] are not applicable for the circle graphs.

2. AN ALGORITHM FOR A MAXIMUM CLIQUE OF A CIRCLE GRAPH

Let G be a given graph with k vertices. We say that G is a *permutation graph* if there exists a permutation $P = [p_1, \dots, p_k]$ of $1, 2, \dots, k$ and a relabeling of the vertices of G by $1, 2, \dots, k$, so that the vertices p_i, p_j are connected if and only if $i > j$ and $p_i < p_j$ in P . Consider a permutation graph G with the corresponding permutation P and the above relabeling of its vertices. Let us direct its edges from low to high. With this orientation, the graph is transitive, since if $p_i \rightarrow p_j$, $p_j \rightarrow p_k$ then $p_i < p_j < p_k$ and $i > j > k$, thus $p_i \rightarrow p_k$. For

example, the permutation corresponding to the permutation graph of Figure 3a is $P = [3, 6, 4, 1, 5, 2]$. For the permutation P we can draw a matching diagram as in Figure 3b; this is called a *permutation diagram*. In this diagram, we write on one side of the dashed line (x, x') the permutation $[1, 2, \dots, k]$, and on the other side the permutation P . We connect the number i of the first row with the number i of the second row. Thus, every segment has one endpoint on each side of (x, x') . It is easy to see that the vertices p_i, p_j of the permutation graph are connected if and only if the segments of p_i and p_j in the diagram, intersect.

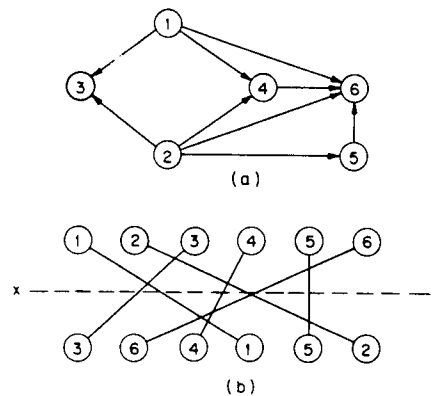


Fig. 3

Consider now, a circle graph G . Let F be its representing family of chords, and denote the chord corresponding to the vertex v by \bar{v} . Denote the subgraph of G on the vertices $\Gamma_v \cup \{v\}$ by G^v . Since we assumed that there are no chords with common endpoints, we can draw a dashed line (x, x') , on the circle, slightly removed from the cord \bar{v} , so that every chord \bar{u} , where $u \in \Gamma_v \cup \{v\}$, has one endpoint on each side of (x, x') . We rename the endpoints of these chords on one side of (x, x') , and the corresponding vertices of G^v , by $1, 2, \dots, k$. Now, the chords form a permutation diagram, so that two chords i and j , $1 \leq i, j \leq k$, intersect if and only if the vertices i and j of G^v are connected. Thus, G^v is a permutation graph.

For example, in Figure 4, the vertices of G^v are v_1, v_2, v_3, v_4, v_5 ; the dashed line was drawn on the circle of Figure 4b; and the chords $\bar{v}_1, \bar{v}_2, \bar{v}_3, \bar{v}_4, \bar{v}_5$ form a permutation diagram of $P = [5, 3, 2, 4, 1]$.

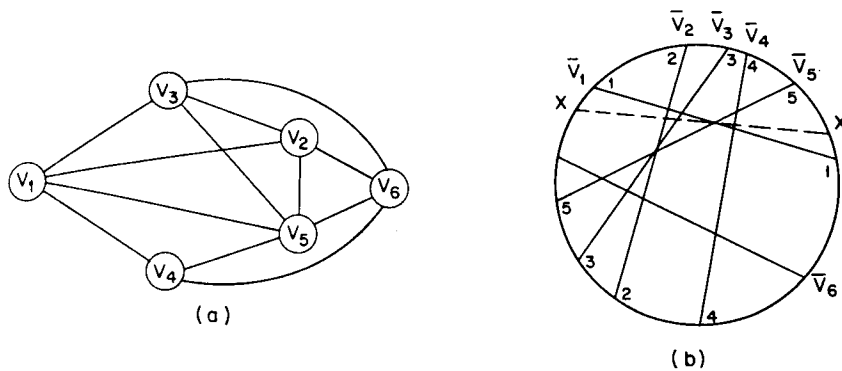


Fig. 4

Therefore, in a circle graph G , for every vertex v , G^v is a permutation graph and thus G^v is transitively orientable.

Let C be a maximum clique of a circle graph G , and let v be any vertex in C . Then the vertices of C are contained in G^v , and therefore C is a maximum clique of G^v . Thus, an algorithm for finding a maximum clique of a circle graph G can work as follows:

- for every vertex v_i of G , consider the subgraph G^{v_i} with its transitive orientation;
- for every $1 \leq i \leq n$, find a maximum clique C^{v_i} of G^{v_i} through the algorithm for transitive graphs, described in the Introduction; a maximum size clique among C^{v_1}, \dots, C^{v_n} is a maximum clique of G .

The graph in Figure 4 has:

$C^{v_1} = C^{v_2} = C^{v_3} = C^{v_5} = \{v_1, v_2, v_3, v_5\}$, $C^{v_4} = \{v_4, v_5, v_6\}$ and $C^{v_6} = \{v_2, v_3, v_5, v_6\}$. Therefore C^{v_1} , for example, is a maximum clique of G . Since finding a maximum clique of a transitive graph takes at most n^2 steps, it follows that the number of steps required by the above algorithm is at most n^3 .

3. AN ALGORITHM FOR A MAXIMUM INDEPENDENT SET OF A CIRCLE GRAPH

Consider a family of intervals on a line. We say that two intervals *overlap*, if they intersect and no one is contained in the other. Let us draw a graph by representing every interval

permutation graph
 permutation P we can
 called a *permu-*
 side of the
 and on the other
 of the first
 every segment
 easy to see that
 connected if and
 intersect.

representing
 to the ver-
 $\{v\}$
 with common
 circle,
 and \bar{u} , where
). We re-
 x, x'), and
 Now, the
 i and j ,
 i and j of

are
 cle of Fig-
 ation diagram

by a vertex, two vertices being connected by an edge if and only if the corresponding intervals overlap. This is called an *overlap graph* (see [5]). Clearly, the overlap graph of the given family is different from its interval graph, since if an interval is contained in another, then, the corresponding vertices are connected in the interval graph, but not in the overlap graph. Consider a family of chords on a circle, for example the one in Figure 5a. We take two opposite points $0, 0'$ on the circle so that 0 is not a chord endpoint, and we draw a tangent at $0'$. Now, we project the chord endpoints from 0 on this tangent. In this way, every chord corresponds to the interval between the projections of its endpoints. Thus we obtain a family of intervals on the tangent, so that two intervals overlap if and only if the corresponding chords intersect. Therefore, the circle graph of the family of chords is isomorphic to the overlap graph of the family of intervals on the tangent. Conversely, if a family of intervals is given on a line, we draw a tangent circle to it and we project the interval endpoints on the circle by a point opposite to the line. Thus, a graph is a *circle graph* if and only if it is an *overlap graph*. Therefore, our problem is reduced to finding an algorithm for a maximum independent set of an overlap graph. As for the families of chords, we can assume here too, without loss of generality, that no two intervals share an endpoint.

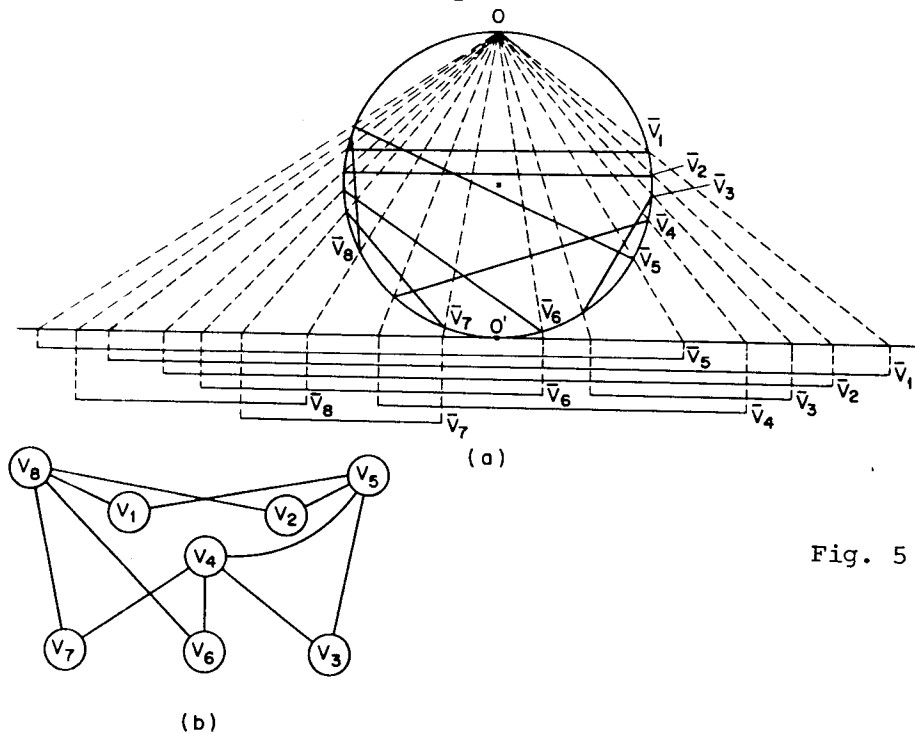


Fig. 5

by α
if H

sum o
is an
we wi
same

inter
respo

let C

we as

tex v

set C
dent

Lemma

dent

Proo

Assu

is a

set

tic

and

G_{u_i}

$\{u_1$

Let us introduce a number of notations. We will denote by $\alpha(H)$ the size of a maximum independent set for a graph H ; if H is a weighted graph, we will denote by $\alpha_w(H)$ the weighted sum of the vertices of a maximum weight independent set. If H is an overlap graph and E its family of representing intervals, we will denote the interval graph of E by \tilde{H} . We will use the same name for corresponding vertices, in H and \tilde{H} .

Consider an overlap graph G , and its family of representing intervals F . For every vertex v of G , we will denote the corresponding interval by \bar{v} . Let us denote also $U_v = \{\bar{u} | \bar{u} \subset \bar{v}\}$, and let G_v be the overlap subgraph with respect to U_v . Assume that we assign a weight $\omega(v)$, where $\omega(v) = \alpha(G_v) + 1$, to every vertex v of \tilde{G} . Let $\{v_1, \dots, v_r\}$ be a maximum weight independent set of \tilde{G} , and for every $1 \leq j \leq r$, let D_j be a maximum independent set of G_{v_j} .

Lemma: $\alpha(G) = \alpha_w(\tilde{G})$ and $\bigcup_{j=1}^r (D_j \cup \{v_j\})$ is a maximum independent set of G .

Proof: Let J be a maximum independent set of G and let

$$A = \{u | u \in J, \text{ and if } \bar{u} \subset \bar{w} \text{ then } w \notin J\}.$$

Assume $A = \{u_1, \dots, u_t\}$. For every $1 \leq i \leq t$, the set

$$J_{u_i} = \{w | w \in J \text{ and } \bar{w} \subset \bar{u}_i\}$$

is an independent set of G_{u_i} . Let C be a maximum independent set of G_{u_i} , and assume $|J_{u_i}| < |C|$.

Then $(J - J_{u_i}) \cup C$ is an independent set of G with more vertices than J , and this is a contradiction. Therefore $|C| = |J_{u_i}|$,

and for every $1 \leq i \leq t$, J_{u_i} is a maximum independent set of

G_{u_i} . Since for every $i \neq j$, $\bar{u}_i \cap \bar{u}_j = \emptyset$, it follows that

$\{u_1, \dots, u_t\}$ is an independent set of \tilde{G} . Therefore

$$\alpha(G) = \sum_{i=1}^t \omega(u_i) \leq \alpha_w(\tilde{G}).$$

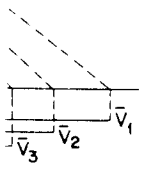


Fig. 5

On the other hand, $\bigcup_{j=1}^r (D_j \cup \{v_j\})$ is an independent set of G ,
 and thus $\alpha_w(\bar{G}) = \sum_{i=1}^r \omega(v_i) = \left| \bigcup_{j=1}^r (D_j \cup \{v_j\}) \right| \leq \alpha(G)$. Therefore
 $\alpha(G) = \alpha_w(\bar{G})$, and $\bigcup_{j=1}^r (D_j \cup \{v_j\})$ is a maximum independent set
 of G .

Q.E.D.

Let us now return to the problem of the algorithm for a maximum independent set of the overlap graph G . Based on the Lemma, we will describe a method for attaching a weight $\omega(v)$ and a set D_v to every v of G , so that $\omega(v) = \alpha(G_v) + 1$, and D_v is a maximum independent set of G_v .

Let us define a family of subsets of F in the following way:

$$\begin{aligned} A_1 &= \{\bar{u} \mid \text{there are no } \bar{v} \in F \text{ such that } \bar{v} \subset \bar{u}\} \\ A_2 &= \{\bar{u} \mid \bar{u} \not\subset A_1, \text{ and if } \bar{v} \subset \bar{u}, \text{ then } \bar{v} \in A_1\} \\ &\vdots \\ A_k &= \{\bar{u} \mid \bar{u} \not\subset \bigcup_{j=1}^{k-1} A_j, \text{ and if } \bar{v} \subset \bar{u}, \text{ then } \bar{v} \in \bigcup_{j=1}^{k-1} A_j\} \\ &\text{and } \bigcup_{i=1}^k A_i = F. \end{aligned}$$

In fact, $\bar{u} \in A_i$ if and only if it contains at least one interval of A_{i-1} , and all the intervals contained in it are elements of $\bigcup_{j=1}^{i-1} A_j$. By the above definition, if $\bar{u}, \bar{v} \in A_i$, then \bar{u} and \bar{v} overlap or they are disjoint. Also, if $i \neq j$, then $A_i \cap A_j = \emptyset$.

Clearly, for every vertex $v \in A_1$, $\omega(v) = 1$ and $D_v = \emptyset$.

Assume by induction on i that all the vertices $u, \bar{u} \in \bigcup_{j=1}^{i-1} A_j$, are attached a weight $\omega(u) = \alpha(G_u) + 1$ and a maximum independent set D_u of G_u . Consider a vertex $v, \bar{v} \in A_i$. For every vertex u of $G_v, \bar{u} \in \bigcup_{j=1}^{i-1} A_j$, and thus, every vertex u of G_v is already

assigned a $\omega(u)$ and D_u . Now, we assign the same weights to the corresponding vertices of \tilde{G}_V (the interval graph of the family of intervals U_V). Since \tilde{G}_V is an interval graph, then as we observed in the Introduction, we can find a maximum weight independent set $\{u_1, \dots, u_t\}$ for it by an efficient algorithm.

Applying the Lemma to the overlap graph G_V , we conclude that

$\alpha(G_V) = \alpha_w(\tilde{G}_V)$ and $\bigcup_{j=1}^t (D_{u_j} \cup \{u_j\})$ is a maximum independent set of G_V . Then, we define $\omega(v) = \alpha(G_V) + 1$ and $D_v = \bigcup_{j=1}^t (D_{u_j} \cup \{u_j\})$.

In this way, we effectively attach a weight $\omega(v) = \alpha(G_V) + 1$ and a maximum independent set D_v of G_V to every vertex v of G . Now,

the algorithm for finding a maximum independent set of G works as follows. Let us assign the weights of the corresponding vertices of G to the vertices of \tilde{G} . Since \tilde{G} is an interval graph, we can find a maximum weight independent set $\{v_1, \dots, v_r\}$ of it.

Then, by the Lemma, $\alpha(G) = \alpha_w(\tilde{G})$ and $\bigcup_{j=1}^r (D_{v_j} \cup \{v_j\})$ is a maximum independent set of G .

Since for finding a maximum weight independent set of \tilde{G}_V we need at most n^2 steps (where n is the number of the vertices of G), the number of steps required for the above algorithm is at most n^3 .

We will apply this algorithm to the overlap graph in Figure 5b, whose family of intervals is described in Figure 5a. For this family of intervals:

$$A_1 = \{\bar{v}_3, \bar{v}_4, \bar{v}_7, \bar{v}_8\}$$

$$A_2 = \{\bar{v}_6\}$$

$$A_3 = \{\bar{v}_2, \bar{v}_5\}$$

$$A_4 = \{\bar{v}_1\}.$$

Therefore:

$$\omega(v_3) = \omega(v_4) = \omega(v_7) = \omega(v_8) = 1;$$

$$D_{v_3} = D_{v_4} = D_{v_7} = D_{v_8} = \phi.$$

ident set of G ,
 $\alpha(G)$. Therefore
 independent set

Q.E.D.
 algorithm for a
 . Based on the
 a weight $\omega(v)$
 $(G_V) + 1$, and

the following

east one inter-
 it are elements

i , then \bar{u} and \bar{v}
 then $A_i \cap A_j = \phi$.
 and $D_v = \phi$.

$i-1$
 $\bar{u}_i \in \bigcup_{j=1}^{i-1} A_j$, are
 an independent
 every vertex u

is already

G_{v_6} has only one vertex, v_7 , and hence $\omega(v_6) = 2$, $D_{v_6} = \{v_7\}$.
 The vertices of G_{v_2} are v_3, v_4, v_6, v_7 , and a maximum weight independent set of \tilde{G}_{v_2} is $\{v_3, v_6\}$, where $\alpha_w(G_{v_2}) = 3$; thus $\omega(v_2) = 4$, and $D_{v_2} = \{v_3, v_6, v_7\}$. The vertices of G_{v_5} are v_6, v_7, v_8 , which form a clique of \tilde{G}_{v_5} . Hence, a maximum weight independent set of \tilde{G}_{v_5} is $\{v_6\}$, where $\alpha_w(\tilde{G}_{v_5}) = 2$, and thus $\omega(v_5) = 3$ and $D_{v_5} = \{v_6, v_7\}$. The vertices of G_{v_1} are $\{v_2, v_3, v_4, v_6, v_7\}$, and a maximum weight independent set of \tilde{G}_{v_1} is $\{v_2\}$, where $\alpha_w(\tilde{G}_{v_1}) = 4$; thus $\omega(v_1) = 5$ and $D_{v_1} = \{v_2, v_3, v_6, v_7\}$. Considering the interval graph \tilde{G} , its unique maximum weight independent set is $\{v_1\}$ when $\alpha_w(\tilde{G}) = 5$; thus $\alpha(G) = 5$ and $\{v_1, v_2, v_3, v_6, v_7\}$ is a maximum independent set of G .

ACKNOWLEDGMENT

The author wishes to express his thanks to Professor Shimon Even for his assistance.

REFERENCES

1. Even, S. and A. Itai, "Queues, Stacks and Graphs," *Theory of Machines and Computations*, Z. Kohavi and A. Paz, ed., Academic Press, New York, 1971, pp. 71-86.
2. Even, S., A. Lempel and A. Pnueli, "Permutation Graphs and Transitive Graphs," *J. ACM*, 19, 1972, pp. 400-410.
3. Dirac, G. A., "On Rigid Circuit Graphs," *Abh. Math.*, University Hamburg, 25, 1961, pp. 71-76.
4. Gilmore, P. C. and A. J. Hoffman, "A Characterization of Comparability Graphs and of Interval Graphs," *Canad. J. Math.*, 16, 1964, pp. 539-548.
5. Fulkerson, D. R. and O. A. Gross, "Incidence Matrices and Interval Graphs," *Pacific J. Math.*, 15, 1965, pp. 835-855.

6. Lekkerkerker, C. G. and J. Ch. Boland, "Representation of a Finite Graph by a Set of Intervals on the Real Line," *Fundamenta Mathematicae*, 51, 1962, pp. 45-64.
7. Gavril, F., "Algorithms for Minimum Coloring, Maximum Clique, Minimum Covering by Cliques, and Maximum Independent Set of a Chordal Graph," *Siam J. Comput.*, 1, 1972, pp. 180-187.

Paper received November 14, 1972.

$D_{v_6} = \{v_7\}$.
 weight inde-
 thus $\omega(v_2) = 4$,
 v_7, v_8 which
 dependent set
 = 3 and
 $\{v_6, v_7\}$, and a
 ere
 $\}$. Considering
 dependent set
 $\{v_3, v_6, v_7\}$ is a

ofessor Shimon
 ," *Theory of*
 , ed.,
 Graphs and
 10.
 ath., Univer-
 zation of
 mad. J. Math,
 rices and
 . 835-855.