ELSEVIER

Discrete Applied Mathematics 69 (1996) 247–255

DISCRETE APPLIED MATHEMATICS

Quasi-threshold graphs ☆

Jing-Ho Yan, Jer-Jeong Chen, Gerard J. Chang *

Department of Applied Mathematics, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 30050, Taiwan

Received 24 January 1994; revised 1 May 1995

Abstract

Quasi-threshold graphs are defined recursively by the following rules: (1) K_1 is a quasi-threshold graph, (2) adding a new vertex adjacent to all vertices of a quasi-threshold graph results in a quasi-threshold graph, (3) the disjoint union of two quasi-threshold graphs is a quasi-threshold graph. This paper gives some new equivalent definitions of a quasi-threshold graph. From them, linear time recognition algorithms follow. We also give linear time algorithms for the edge domination problem and the bandwidth problem in this class of graphs.

1. Introduction

Many well-known classes of graphs can be obtained from a vertex by recursively applying one or more graph operations. As an example, a tree can be obtained from a vertex by recursively adding a new vertex that is adjacent to exactly one old vertex. Some graph operations commonly used for generating graphs from a vertex are:

(o_1) adding a new isolated vertex;

(o_2) adding a new vertex that is adjacent to all old vertices;

(o_3) adding a new vertex that is adjacent to exactly one old vertex;

(o_4) adding a new vertex that is adjacent to a clique;

(o_5) adding a new vertex v' that is adjacent to all neighbors of an old vertex v;

(o_6) adding a new vertex v' that is adjacent to an old vertex v and all its neighbors;

(o_7) graph complement;

(o_8) disjoint union of two graphs;

(o_9) join of two graphs.

It is well known that operations (o_1) and (o_2) produce threshold graphs (see [4]), operation (o_3) produces trees, operations (o_1) and (o_3) produce forests, operations (o_1) and (o_4) produce chordal graphs (see [7, 10]), operations (o_7) and (o_8) (or (o_8) and (o_9)) produce cographs (see [5]), and operations (o_1), (o_3), (o_5), and (o_6) produce

☆ Supported in part by the National Science Council under grant NSC81-0208-M009-26.

* Corresponding author. E-mail: gjchang@math.nctu.edu.tw

SSDI 0166-218X(96)00094-1

distance-hereditary graphs (see [1]). This paper studies the class of graphs produced by operations (o_1), (o_2), and (o_8) (or (o_2) and (o_8), since (o_1) is a special case of (o_8)). Wolk [14] called these graphs *comparability graphs of trees* and gave characterizations of them. Golumbic [9] called them *trivially perfect* graphs in respect to a certain concept of "perfection." Ma et al. [12] called them *quasi-threshold* graphs and studied algorithmic results. In particular, they gave an $O(|V||E|)$ time algorithm for the recognition problem, an $O(|V|^2)$ time algorithm for the bandwidth problem, and a polynomial time algorithm for the Hamiltonian cycle problem.

The main purpose of this paper is to characterize quasi-threshold graphs in further detail. From the characterizations identified here, linear time recognition algorithms follow. We also give linear time algorithms for the edge domination problem and the bandwidth problem in this class of graphs.

2. Characterizations

A graph is *H-free* if it does not contain H as an induced subgraph.

A subclass of *quasi-threshold* graphs, the threshold graphs, was introduced by Chvátal and Hammer [4], who described many properties of threshold graphs. A forbidden subgraph characterization of these graphs is as follows.

Theorem 1 (Chvátal and Hammer[4]). *A graph is a threshold graph if and only if it is P_4-free, C_4-free, and $2K_2$-free.*

Cographs, a super class of quasi-threshold graphs, have also been extensively studied. The following is a well-known forbidden subgraph characterization of cographs.

Theorem 2 (Corneil et al.[5]). *A graph is a cograph if and only if it is P_4-free.*

A graph is *chordal* (or *triangulated*) if every cycle of length greater than three has at least one *chord*, which is an edge joining two non-consecutive vertices in the cycle. In other words, a graph is chordal if it is C_n-free for all $n \geqslant 4$. Chordal graphs have been extensively studied from the perspective of perfect graph theory (see [10]).

Suppose $\mathscr{F} = \{S_x \mid x \in V\}$ is a family of sets. The *intersection graph* of $\mathscr{F}$ is the graph whose vertex set is V and two distinct vertices x and y are adjacent if and only if $S_x \cap S_y \neq \emptyset$. It is well known that a graph is chordal if and only if it is the intersection graph of a family of subtrees of a tree (see [10]). An *interval graph* is the intersection graph of a family of intervals in the real line. Interval graphs are chordal graphs.

In a graph $G = (V,E)$, the *neighborhood* of a vertex v is $N(v) = \{u \in V \mid uv \in E\}$ and the *closed neighborhood* of v is $N[v] = \{v\} \cup N(v)$. The *degree* $deg(v)$ of v is $|N(v)|$. A *clique* (respectively, *stable set*) is a set of pairwise adjacent (respectively, non-adjacent) vertices. Let $m(G)$ denote the number of maximal cliques of a graph G and let $\alpha(G)$ be the maximum size of a stable set in G. It is clear that

$$\alpha(G) \leqslant m(G)$$

since there must be $\alpha(G)$ distinct maximal cliques containing the vertices of a maximum stable set.

A *rooted tree* is a directed graph obtained from a tree by assigning each edge a direction so that there exists a special vertex r, called the *root*, such that there is a unique directed path from r to each vertex. A *rooted forest* is the disjoint union of several rooted trees. The *induced graph* of a rooted forest $F = (V,E)$ is the graph $G(F) = (V,E')$, where $uv \in E'$ if and only if $u \neq v$ and there is a directed u–v or v–u path in F. F is called a *rooted forest representation* of $G(F)$.

Let D be the *transitive closure* of a rooted forest F, i.e., D has the same vertex set as F and uv is an arc in D if and only if $u \neq v$ and there is a directed u–v path in F. D can be regarded as a poset in which $x > y$ if and only if xy is an arc. $G(F)$ is then the comparability graph of the poset D. Note that F is the Hasse diagram of D.

Now we are ready to state characterizations of quasi-threshold graphs.

Theorem 3. *The following statements are equivalent for any graph G.*

(1) *G is a quasi-threshold graph.*

(2) *G is a cograph and is an interval graph.*

(3) *G is a cograph and is a chordal graph.*

(4) *G is P_4-free and C_4-free.*

(5) *For any edge uv in G, either $N[u] \subseteq N[v]$ or $N[v] \subseteq N[u]$.*

(6) *If $v_1, v_2, \ldots, v_n$ is a path with $deg(v_1) \geqslant deg(v_2) \geqslant \cdots \geqslant deg(v_{n-1})$, then $\{v_1, v_2, \ldots, v_n\}$ is a clique.*

(7) *G is induced by a rooted forest.*

(8) *$\alpha(H) = m(H)$ for all induced subgraphs H of G.*

Proof. The proof that (4) is equivalent to (8) can be found in [9]. The proof that (4) is equivalent to (7) can be found in [15]. We shall prove (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (4) $\Rightarrow$ (5) $\Rightarrow$ (6) $\Rightarrow$ (7) $\Rightarrow$ (1).

(1) $\Rightarrow$ (2): Note that in the definitions of quasi-threshold graphs and cographs, operation (o_2) is the same as applying operation (o_7), followed by (o_1), and then (o_7). Thus a quasi-threshold graph is a cograph. We shall prove that a quasi-threshold graph is an interval graph by induction. First, K_1 is the intersection graph of $\{[0, 1]\}$. Suppose G is obtained from a quasi-threshold graph H by applying operation (o_2). By the induction hypothesis, H is the intersection graph of a family $\mathscr{F}$ of intervals. Let I^* be an interval that intersects all intervals in $\mathscr{F}$. Then G is the intersection graph of $\mathscr{F} \cup \{I^*\}$. Suppose G is the union of two quasi-threshold graphs G_1 and G_2. By the induction hypothesis, G_1 (respectively, G_2) is the intersection graph of a family $\mathscr{F}_1$ (respectively, $\mathscr{F}_2$) of intervals. We may assume that no interval in $\mathscr{F}_1$ intersects an interval in $\mathscr{F}_2$, otherwise we can shift all intervals in $\mathscr{F}_2$ to the right by a large unit.

Then G is the intersection graph of $\mathscr{F}_1 \cup \mathscr{F}_2$.

(2) $\Rightarrow$ (3): This holds because an interval graph is chordal.

(3) $\Rightarrow$ (4): This holds because a cograph is P_4-free by Theorem 2 and a chordal graph is C_4-free by the definition of a chordal graph.

(4) $\Rightarrow$ (5): Suppose uv is an edge such that $N[u] \not\subseteq N[v]$ and $N[v] \not\subseteq N[u]$. Then there is a vertex u' adjacent to u but not v and a vertex v' adjacent to v but not u. Thus, $\{u', u, v, v'\}$ induces a P_4 or C_4, which is impossible.

(5) $\Rightarrow$ (6): For any $1 \leqslant i \leqslant n-1$, $v_i v_{i+1}$ is an edge. By (5) and the assumption $deg(v_i) \geqslant deg(v_{i+1})$, $N[v_{i+1}] \subseteq N[v_i]$ for $1 \leqslant i \leqslant n-2$. Then, for any $1 \leqslant i < j \leqslant n$, $v_j \in N[v_{j-1}] \subseteq N[v_i]$ and so $v_i v_j \in E$. Therefore, $\{v_1, v_2, \ldots, v_n\}$ is a clique.

(6) $\Rightarrow$ (7): Suppose we label (arbitrarily) the vertices of G as $1, 2, \ldots, |V|$. For each edge ij, direct the edge from i to j if $deg(i) > deg(j)$ or $deg(i) = deg(j)$ with $i < j$. This results in an acyclic directed graph D. For two arcs ij and jk in D, i, j, k is a path in G and $deg(i) \geqslant deg(j) \geqslant deg(k)$. By (6), ik is an edge in G. For the case of $deg(i) > deg(k)$, ik is an arc in D. For the case of $deg(i) = deg(j) = deg(k)$, by the edge orientation rule, $i < j < k$ and so ik is an arc in D. Therefore D is a transitive directed graph, i.e., D defines a poset whose Hasse diagram is F. We claim that F is a rooted forest. If this is not the case, then there exist arcs ij and kj in F such that ik is not an edge in G. By the edge orientation rule, i, j, k is a path in G and $deg(i) \geqslant deg(j)$. By (6), ik is an edge, which is impossible. Therefore G is induced by the rooted forest F.

(7) $\Rightarrow$ (1): Suppose G is induced by a rooted forest F of n vertices. We shall prove that G is a quasi-threshold graph by induction on n. The case of $n = 1$ is clear. Suppose the assertion is true for all $n' < n$. For the case where F is the union of two rooted forests F_1 and F_2, G is the union of $G(F_1)$ and $G(F_2)$. By the induction hypothesis, $G(F_1)$ and $G(F_2)$ are quasi-threshold graphs. Then G is obtained from $G(F_1)$ and $G(F_2)$ by operation (o_8), and G is a quasi-threshold graph. For the case where F is a rooted tree with root r, $F - r$ is a rooted forest and so by the induction hypothesis, $G(F - r)$ is a quasi-threshold graph. G is obtained from $G(F - r)$ by applying operation (o_2). So G is again a quasi-threshold graph. □

The above characterizations provide several linear time algorithms for recognizing quasi-threshold graphs. First, the well-known linear time algorithms [2, 6, 11, 13] for recognizing cographs, interval graphs, and chordal graphs do the job. Although these algorithms are linear, they are quite complicated.

The proof of (6) $\Rightarrow$ (7) provides a much simpler way to recognize a quasi-threshold graph. This method also produces a rooted forest representation if the graph is a quasi-threshold graph. The method is as follows. First, produce the acyclic digraph D as in the proof of (6) $\Rightarrow$ (7). For each vertex j in D with $indegree(j) \geqslant 1$, choose an arc ij of D such that $indegree(i)$ is largest. All vertices of D together with all of these arcs form a rooted forest F, which is a spanning subdigraph of D. Check whether the transitive closure of F is D. This can be done by examining if the number of ancestors of j in F is equal to the indegree of j for each vertex j in D. If D is the transitive closure of F, then G is a quasi-threshold graph and F is a rooted forest representation of G. Otherwise, G is not a quasi-threshold graph.

More precisely, we have the following algorithm. Note that we do not need to create the digraph D, because the degrees of all vertices in G determine D.

Algorithm QT. *Test whether a graph is a quasi-threshold graph.*

Input: *A graph* $G = (V,E)$ *with* $V = \{v_1, v_2, \ldots, v_n\}$.

Output: *A rooted forest representation F of G if G is a quasi-threshold graph. Otherwise output "no."*

Method.

calculate $deg(v_i)$ for each vertex v_i in G;
for $i = 1$ **to** n **do** $indegree(v_i) \longleftarrow 0$;
for each edge $\{v_i, v_j\}$ in G **do**
    if $deg(v_i) > deg(v_j)$ **or** $(deg(v_i) = deg(v_j)$ **and** $i < j)$
    then $indegree(v_j) \longleftarrow indegree(v_j) + 1$
    else $indegree(v_i) \longleftarrow indegree(v_i) + 1$;
$F \longleftarrow \emptyset$; /* a digraph with vertex set $\{v_1, v_2, \ldots, v_n\}$ and no edges */
for $j = 1$ **to** n **do**
    if $indegree(j) \geqslant 1$
    then choose a vertex $v_i \in N_G[v_j]$ such that $deg(v_i) > deg(v_j)$ or
        $(deg(v_i) = deg(v_j)$ and $i < j)$ and indegree(i) is largest;
    add the arc (i, v_j) into F;
    end do;
use a depth-first search to compute the number $anc(v_j)$ of ancestors of v_j in F for each j;
if $indegree(v_j) = anc(v_j)$ for all j **then** output F **else** answer "no".

3. Edge domination

This section presents a linear time algorithm for the edge domination problem in quasi-threshold graphs. Suppose G is a quasi-threshold graph and F is a rooted forest representation of G.

An *edge dominating set* of a graph G is a set of edges D such that every edge not in D is adjacent to some edge in D. The *edge domination problem* is to find the *edge domination number* $\gamma_e(G)$ of G, which is the minimum size of an edge dominating set in G.

Our solution to the edge domination problem in quasi-threshold graphs is through the following problem. An *edge cover* of a graph G is a set of edges such that every vertex in G is incident to some edge in C. The *edge covering number* $c_e(G)$ of G is the minimum size of an edge cover of G. For convenience, if G has isolated vertices, we include all isolated vertices in any edge cover C so that each isolated vertex is "covered" by itself in C.

Theorem 4. *Suppose G is a quasi-threshold graph and F a rooted forest representation of G. Let F' be the rooted forest obtained from F by deleting all leaves, i.e., vertices of outdegree zero. If F' induces G', then $\gamma_e(G) = c_e(G')$.*

Proof. We may assume that G is connected and so F is a rooted tree. If F has just one vertex, then $G' = \emptyset$ and so $\gamma_e(G) = c_e(G') = 0$. If F is a star of at least two vertices, then F' has just one vertex and so $\gamma_e(G) = c_e(G') = 1$. Now suppose F is not a star.

Let D be a minimum edge dominating set of G. For any directed path P from the root r to a leaf v in F, there is at most one vertex which is not incident to any edge in D. Suppose there is some edge $vw \in D$. Then w is in P. Let $V(P)$ be the set of vertices of P and $V(D)$ the set of vertices of G incident to some edge of D. For the case where $V(P) - V(D) = \{u\}$, $D - vw + uw$ is a minimum edge dominating set of G. For the case where $V(P) = V(D)$ has a vertex u different from v or w, $D - vw + uw$ is also a minimum edge dominating set of G. For the case of $V(P) = V(D) = \{w, v\} = \{r, v\}$, $D - vw + uw$ is a minimum edge dominating set of G, where u is a non-leaf vertex adjacent to r in F. In any case, we may assume that v is the only vertex of P that is not incident to any edge in D. Hence $V(D) = V(F')$, i.e., D is an edge cover of G'. This proves that $\gamma_e(G) \geqslant c_e(G')$.

On the other hand, any edge cover of G' covers all vertices in G'. This together with the fact that all leaves of F form a stable set in G implies $\gamma_e(G) \leqslant c_e(G')$. So $\gamma_e(G) = c_e(G')$. □

Theorem 4 transforms the edge domination problem in G into the edge cover problem in G'. We can in turn transform this problem into the matching problem. A *matching* is a set of pairwise non-adjacent edges. Denote by $m_e(G)$ the maximum size of a matching in G. Then we have following theorem.

Theorem 5 (Gallai [8]). $c_e(G) + m_e(G) = |V(G)|$ *for any graph G.*

Furthermore, suppose M^* is a maximum matching of G. For any vertex x not incident to any edge in M^*, choose an edge e_x incident to x. Then these edges together with M^* form a minimum edge cover of G. So the edge domination problem in G is equivalent to the maximum matching problem in G', which is also quasi-threshold. The following two lemmas are easy and provide a linear time algorithm for finding a maximum matching of a quasi-threshold graph. Proofs of the lemmas and their implementation to an algorithm are obvious and hence omitted.

Lemma 6. $m_e(G \cup H) = m_e(G) + m_e(H)$ *for any two disjoint G and H.*

Lemma 7. *If G is the graph obtained from a graph H by adding a new vertex v adjacent to all vertices in H, then*

$$m_e(G) = \begin{cases} m_e(H), & \text{if } 2m_e(H) = |V(H)|, \\ m_e(H) + 1, & \text{if } 2m_e(H) < |V(H)|. \end{cases}$$

4. Bandwidth

A *labeling* of a graph $G = (V, E)$ is a bijection from V to $\{1, 2, \ldots, |V|\}$. The *bandwidth of G with respect to a labeling f* is defined to be $B(G, f) = \max_{uv \in E} |f(u) - f(v)|$. The *bandwidth* of G is $B(G) = \min B(G, f)$, where the minimum is taken over all labelings f of G. A labeling f of G is *bandwidth optimal* if $B(G, f) = B(G)$. For a survey of the bandwidth problem, see [3].

Lemma 8. *If G is a subgraph of H, then $B(G) \leqslant B(H)$.*

Lemma 9. *$B(G \cup H) = \max\{B(G), B(H)\}$ for any two disjoint graphs G and H.*

Lemma 10. *If Δ is the maximum degree of a graph G, then $B(G) \geqslant \lceil \Delta/2 \rceil$.*

Theorem 11. *Suppose $n_1 \geqslant \cdots \geqslant n_r$ and $m_i = \sum_{j=1}^{i} n_j$ for $1 \leqslant i \leqslant r$. Consider r disjoint graphs $G_1 = (V_1, E_1), \ldots, G_r = (V_r, E_r)$ of order $n_1, \ldots, n_r$ respectively, where $V_i = \{v_{m_{i-1}+1}, \ldots, v_{m_i}\}$ for $1 \leqslant i \leqslant r$. Let f be a labeling of G_1, with $f(v_j) = j$ for $1 \leqslant j \leqslant n_1$. Let $G = (V, E)$ be the graph obtained from $\bigcup_{1 \leqslant i \leqslant r} G_i$ by adding a new vertex v_0 adjacent to all v_j with $1 \leqslant j \leqslant m_r$. Consider the following labeling g of G:*

$$g(v_j) = \begin{cases} j, & \text{if } 1 \leqslant j \leqslant \lceil m_r/2 \rceil, \\ \lceil m_r/2 \rceil + 1, & \text{if } j = 0, \\ j + 1, & \text{if } \lceil m_r/2 \rceil + 1 \leqslant j \leqslant m_r \end{cases}$$

If $n_1 \leqslant \lceil m_r/2 \rceil$, then g is an optimal bandwidth labeling of G. If $n_1 \geqslant \lceil m_r/2 \rceil + 1$ and f is an optimal bandwidth labeling of G_1, then g is an optimal bandwidth labeling of G.

Proof.

$$\begin{aligned} B(G, g) &= \max_{v_j v_k \in E} |g(v_j) - g(v_k)| \\ &= \max\{\max_{1 \leqslant k \leqslant m_r} |g(v_0) - g(v_k)|, \max_{1 \leqslant i \leqslant r} \max_{v_j v_k \in E_i} |g(v_j) - g(v_k)|\} \\ &= \max\{\lceil m_r/2 \rceil, \max_{1 \leqslant i \leqslant r} \max_{v_j v_k \in E_i} |g(v_j) - g(v_k)|\} \end{aligned}$$

Since $n_1 \geqslant \cdots \geqslant n_r$, $\lceil m_r/2 \rceil \geqslant n_2 \geqslant \cdots \geqslant n_r$. For every i with $2 \leqslant i \leqslant r$ and $v_j \in V_i$, we have $m_i + 1 \leqslant g(v_j) \leqslant m_i + 1$. Therefore $\max_{v_j v_k \in E_i} |g(v_j) - g(v_k)| \leqslant n_i \leqslant \lceil m_r/2 \rceil$. Thus

$$B(G, g) = \max\{\lceil m_r/2 \rceil, \max_{v_j v_k \in E_1} |g(v_j) - g(v_k)|\}.$$

For the case of $n_1 \leqslant \lceil m_r/2 \rceil$, by the same argument as above, $\max_{v_j v_k \in E_1} |g(v_j) - g(v_k)| \leqslant \lceil m_r/2 \rceil$ and so $B(G, g) = \lceil m_r/2 \rceil$. This together with Lemma 10 implies that g is an optimal bandwidth labeling of G.

Next we consider the case where $n_1 \geqslant \lceil m_r/2 \rceil + 1$ and f is an optimal bandwidth labeling of G_1. Note that $\max_{v_j v_k \in E_1} |g(v_j) - g(v_k)| = B(G_1)$ or $B(G_1) + 1$. Hence $B(G,g) = \max\{\lceil m_r/2 \rceil, B(G_1)$ or $B(G_1) + 1\}$. For the case of $B(G,g) = \lceil m_r/2 \rceil$, by Lemma 10, g is an optimal bandwidth labeling of G. For the case of $B(G,g) = B(G_1)$, by Lemma 8, $B(G,g) \leqslant B(G_1) \leqslant B(G)$ and so g is an optimal bandwidth of G. So we may assume that $B(G,g) = B(G_1) + 1 > \lceil m_r/2 \rceil$. Suppose g^* is an optimal bandwidth labeling of G. For any edge $xy \in E$ with $g^*(x) < g^*(y)$ and $|g^*(x) - g^*(y)| = B(G)$, it is the case that $g^*(x) \leqslant g^*(v_0) \leqslant g^*(y)$. Otherwise, either $1 \leqslant g^*(x) < g^*(y) < g^*(v_0)$ or $g^*(v_0) < g^*(x) < g^*(y) \leqslant 1 + m_r$ would imply $|g^*(x) - g^*(v_0)| > |g^*(x) - g^*(y)| = B(G)$ or $|g^*(v_0) - g^*(y)| > |g^*(x) - g^*(y)| = B(G)$, in contradiction to $xv_0 \in E$ and $v_0 y \in E$. Consider the labeling g' of $G - v_0$ defined by

$$g'(x) = \begin{cases} g^*(x) & \text{if } g^*(x) < g^*(v_0), \\ g^*(x) - 1 & \text{if } g^*(x) > g^*(v_0). \end{cases}$$

Then $B(G - v_0, g') \leqslant B(G) - 1$ and so

$$B(G,g) = B(G_1) + 1 \leqslant B(G - v_0) + 1 \leqslant B(G - v_0, g') + 1 \leqslant B(G).$$

Therefore, g is an optimal bandwidth labeling of G. □

To compute the bandwidth of a graph, by Lemma 9, we may assume the graph is connected. Suppose G is a connected quasi-threshold graph with a rooted tree representation T rooted at w. For any vertex v of T, let T_v be the subtree of T rooted at v and let $des(v)$ the number of vertices of T_v. T_v induces a subgraph G_v of G with $des(v)$ vertices. If a vertex v_0 has r children $v_1, v_2, \ldots, v_r$ with $des(v_1) \geqslant des(v_2) \geqslant \cdots \geqslant des(v_r)$ in T, then G_{v_0} is precisely the graph obtained from $\bigcup_{1 \leqslant i \leqslant r} G_{v_i}$ by adding a new vertex v_0 adjacent to all other vertices. So the solution to G_{v_1} provides a solution to G_{v_0} by Theorem 11. A standard depth-first search with some modifications provides an efficient implementation. We omit the algorithm as it is not hard.

Acknowledgements

The authors thank two anonymous referees for many constructive suggestions for a revision of this paper.

References

[1] H.J. Bandelt and H.M. Mulder, Distance-hereditary graphs, J. Combin. Theory, Ser. B 41 (1986) 182–208.

[2] K.S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithm, J. Comput System. Sci. 13 (1976) 335–379.

[3] P.Z. Chinn, J. Chvatalova, A.K. Dewdney and N.E. Glbbs, The bandwidth problem for graphs and matrices – a survey, J. Graph Theory 6 (1982) 223–254.

[4] V. Chvátal and P.L. Hammer, Set-packing and threshold graphs, Res. Report CORR 73–21, University of Waterloo (1973).

[5] D.G. Corneil, H. Lerchs and L. Stewart Burlingham, Complement reducible graphs, Discrete. Appl. Math. 3 (1981) 163–174.

[6] D.G. Corneil, Y. Perl and L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (1985) 926–934.

[7] P. Duchet, Classical perfect graphs, Ann. Discrete Math. 21 (1984) 67–101.

[8] T. Gallai, Über extreme punkt- und kantenmengen, Ann. Univ. Sci. Budapest. Eötvös Sect. Math. 2 (1959) 133–138.

[9] M.C. Golumbic, Trivially perfect graphs, Discrete Math. 24 (1978) 105–107.

[10] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, (Academic Press, New York, 1980).

[11] W.L. Hsu, A simple test for interval graphs, Lecture Notes in Computer Science 657 (1992) 11–16.

[12] S. Ma, W.D. Wallis and J. Wu, Optimization problems on quasi-threshold graphs, J. Combin. Inform. System. Sci. 14 (1989) 105–110.

[13] D.J. Rose, R.E. Tarjan and G.S. Leuker, Algorithmic aspects of vertex elimination on graphs, SIAM. J. Comput. 5 (1976) 266–283.

[14] E.S. Wolk, The comparability graph of a tree, Proc. Amer. Math. Soc. 3 (1962) 789–795.

[15] E.S. Wolk, A note on the comparability graph of a tree, Proc. Amer. Math. Soc. 16 (1965) 17–20.