Threshold Tolerance Graphs

Clyde L. Monma
BELL COMMUNICATIONS RESEARCH
MORRISTOWN, NEW JERSEY
Bruce Reed
UNIVERSITY OF WATERLOO
WATERLOO, ONTARIO, CANADA
William T. Trotter, Jr.
UNIVERSITY OF ARIZONA
TUSCON, ARIZONA

ABSTRACT

In this paper, we introduce a class of graphs that generalize threshold graphs by introducing threshold tolerances. Several characterizations of these graphs are presented, one of which leads to a polynomial-time recognition algorithm. It is also shown that the complements of these graphs contain interval graphs and threshold graphs, and are contained in the subclass of chordal graphs called strongly chordal graphs, and in the class of interval tolerance graphs.

1. INTRODUCTION

An undirected graph $G = (V, E)$ is called a *threshold tolerance graph* if it is possible to associate weights and tolerances with each vertex of G so that two vertices are adjacent exactly when the sum of their weights exceeds either of their tolerances. More formally, there are weights w_v and tolerances t_v for each $v \in V$ so that

$$xy \in E \langle = \rangle\ w_x + w_y \geq \min(t_x, t_y). \qquad (*)$$

If we insist that all tolerances be equal, we obtain the class of *threshold graphs* [4]; see also [12; 13, chap. 10; 17; 21]. It is easy to see that we may require that all weights and tolerances are positive, and that strict inequality holds in (*).

Threshold tolerance graphs are interesting because they generalize threshold graphs. The complements of threshold tolerance graphs, which we call coTT

Journal of Graph Theory, Vol. 12, No. 3, 343–362 (1988)
 CCC 0364-9024/88/030343-20$04.00

graphs, are also interesting. This class includes not only all threshold graphs (since the complement of a threshold graph is a threshold graph) but is also related to other well-studied classes of graphs, as shown in Theorem 1.1 below. To prove this theorem we will need the following alternate definition of coTT graphs. A graph $G = (V, E)$ is a *coTT graph* if there are numbers a_v and b_v for every $v \in V$ so that

$$xy \in E \Leftrightarrow a_x \leq b_y \quad \text{and} \quad a_y \leq b_x .$$

To see that these are precisely the complements of threshold tolerance graphs, set $a_x = w_x$ and $b_x = t_x - w_x$. As before, we may take all of these numbers to be positive.

A graph $G = (V, E)$ is called an *interval graph* [2; 9; 11; 13, chap. 8; 18] if there are closed intervals $I_v = [L_v, R_v]$ of the real line for each $v \in V$ so that two vertices are adjacent exactly when their intervals intersect, that is,

$$xy \in E \Leftrightarrow I_x \cap I_y \neq \emptyset .$$

A graph $G = (V, E)$ is called an *interval tolerance graph* [14, 15] if there are intervals $I_v = [L_v, R_v]$ and tolerances τ_v for each $v \in V$ so that

$$xy \in E \Leftrightarrow |I_x \cap I_y| \geq \min(\tau_x, \tau_y)$$

where $|I|$ is the length of interval I. A graph $G = (V, E)$ is called a *chordal graph* [3, 6, 9, 10, 16, 18, 22, 23] if it contains no induced chordless cycle C_n of length $n \geq 4$. We let P_n denote a path on n vertices and K_n denote the complete graph on n vertices.

Theorem 1.1.

(a) Every threshold graph is a coTT graph.
(b) Every interval graph is a coTT graph.
(c) Every coTT graph is an interval tolerance graph.

Proof. Let $G = (V, E)$ be a threshold graph with representation by weights w_v for every $v \in V$ and threshold t. Define $a_v = -w_v$ and $b_v = w_v - t$ to obtain a coTT representation for G since $w_x + w_y \geq t$ if and only if $a_x \leq b_y$ and $a_y \leq b_x$. [Note that $(b) \Rightarrow (a)$ since every threshold graph is an interval graph.]

Let $G = (V, E)$ be an interval graph with representation by intervals $I_v = [L_v, R_v]$ for every $v \in V$. Define $a_v = L_v$ and $b_v = R_v$ to obtain a coTT representation for G since $I_x \cap I_y \neq \emptyset$ if and only if $a_x \leq b_y$ and $a_y \leq b_x$.

Let $G = (V, E)$ be a coTT graph with representation by a_v and b_v for every $v \in V$. As stated previously, we may take all values to be positive. Define $I_v = [L_v, R_v] = [a_v, a_v + b_v]$ and $\tau_v = a_v$ to obtain an interval tolerance representation for G since $|I_x \cap I_y| \geq \min(\tau_x, \tau_y)$ if and only if $a_x \leq b_y$ and $a_y \leq b_x$. ∎

The example graphs in Figure 1 show that the containments in Theorem 1.1 are all strict.

In section 2, we obtain a characterization of coTT graphs. We also show that coTT graphs are contained in the subclass of chordal graphs called strongly chordal graphs [8] (also called sun-free [5] graphs). In section 3, we present alternate characterizations of coTT graphs, one of which leads to a polynomial-time algorithm for recognizing coTT graphs. Concluding remarks and open problems are presented in section 4.

2. CHARACTERIZATION

Before presenting the characterizations of coTT graphs, we first make a few definitions. We say that x *sees* y in $G = (V, E)$ if $xy \in E$; otherwise, we say that x *misses* y. An *independent set* is a set of vertices where each vertex misses every other. A *clique* is a set of vertices where each vertex sees every other.

The *neighborhood* $N(v)$ of a vertex v in $G = (V, E)$ is given by the set of vertices which v sees. The *closed neighborhood* $\hat{N}(v)$ of v is given by v together with its neighborhood. A vertex v in G is called *simplicial* if $N(x)$ is a clique in G. Two vertices x and y in G are *compatible* if $\hat{N}(x) \subseteq \hat{N}(y)$ or vice versa. A vertex v in G is *simple* if the vertices in $N(v)$ are pairwise compatible. We note that a simple vertex is simplicial.

A graph G is called *strongly chordal* [8] if every induced subgraph has a simple vertex. A similar characterization holds for chordal graphs.

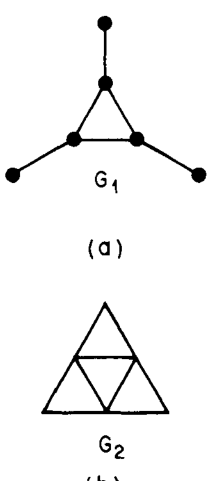

FIGURE 1. Example graphs. (a) G_1 is coTT but not interval or threshold. (b) G_2 is interval but not coTT.

Theorem 2.1 [6, 18]. A graph G is chordal if and only if every induced subgraph of G has a simplicial vertex. ∎

Chordal graphs were originally defined in terms of forbidden subgraphs, i.e., no C_n for $n \geq 4$. Farber [8] obtains a forbidden subgraph characterization for strongly chordal graphs. A *trampoline* is a graph $G = (V, E)$ on $2k$ vertices for some $k \geq 3$ whose vertices can be partitioned into $W = \{w_1, w_2, \ldots, w_k\}$ and $U = \{u_1, u_2, \ldots, u_k\}$ so that W is independent, U forms a clique, and w_i is adjacent to u_j if and only if $i = j$ or $i = j + 1 (\mathrm{mod}\ k)$. Figure 1(b) is a trampoline with $k = 3$.

Theorem 2.2 [8]. A chordal graph G is strongly chordal if and only if G contains no induced trampoline. ∎

In order to show that all coTT graphs are strongly chordal, we will need to characterize both classes in terms of special types of orders on the vertices. We will use the symbol $<$ to denote a partial order on the vertices. We say that x *precedes* y in the order if $x < y$; in this case we also say that y *follows* x in the order. A vertex x that has no other vertex preceding it in the order is called *initial*. We extend this order to sets of vertices S and T so that $S < T$ means $x < y$ for every $x \in S$ and $y \in T$.

An *elimination ordering* [22] of $G = (V, E)$ is a total ordering $<$ of V so that for all $v \in V$, $\{w \in N(v) : v < w\}$ induces a complete graph in G; i.e., v is simplicial in the subgraph induced by v and the vertices following v in the order. A *simple elimination ordering* [8] of $G = (V, E)$ is a total ordering $<$ of V so that for all $v \in V$, the vertices of $\{w \in N(v) : v < w\}$ are pairwise compatible; i.e., v is simple in the subgraph induced by v and the vertices following v in the order. A *strong elimination ordering* [8] of $G = (V, E)$ is a total ordering of V in which neither of the two ordered induced subgraphs shown in Figure 2(a and b) occur. (The order in Figure 2 is given by $w < x < y < z$.) We note that an elimination ordering forbids exactly the induced subgraph shown in Figure 2(a).

Theorem 2.3 [9, 22]. A graph G is chordal if and only if G has an elimination ordering. Any simplicial vertex may start the elimination ordering. ∎

Theorem 2.4 [8]. A graph G is strongly chordal if and only if G has a simple elimination ordering. Any simple vertex may start the simple elimination ordering. Furthermore, a graph G is strongly chordal if and only if G has a strong elimination ordering. ∎

It is not hard to see that if there is a strong elimination ordering of the vertices in a graph, then each vertex v is simple in the subgraph induced by v and the vertices following v in the order. Thus, if G permits a strong elimination order then G is strongly chordal. It is not evident that every strongly chordal

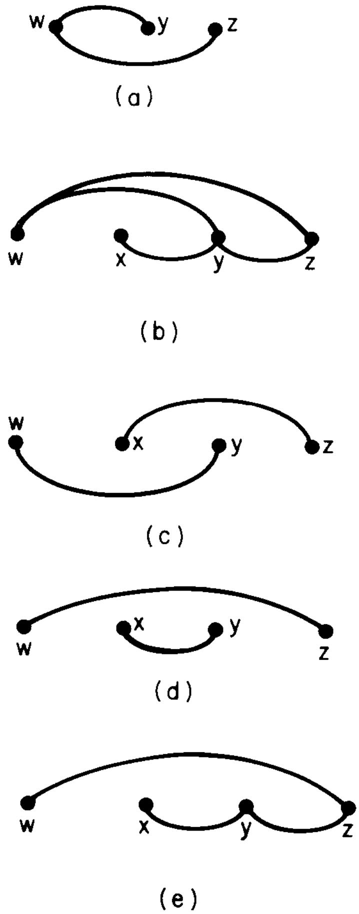

FIGURE 2. Forbidden configurations in proper orders where $w < x < y < z$.

graph permits a strong elimination order. Farber developed an algorithm that, given a strongly chordal graph G, finds a strong elimination order on G. We describe the algorithm below.

Algorithm 1 (Farber's Algorithm)

Input: A graph $G = (V, E)$.
Output: A strong elimination order $<$ if G is strongly chordal, or an induced subgraph with no simple vertex otherwise.

At any given time, we will have a subgraph H of vertices of G whose position in the order has not been determined. We know that $G - H$ preceeds H and we have a total order on $G - H$.

Step 0: Set $H \leftarrow G, U \leftarrow \emptyset, < \leftarrow \emptyset$.
Step 1: For each pair of adjacent vertices x and y in H, if $\hat{N}_H(x) \subset \hat{N}_H(y)$ then set $U \leftarrow U + \overrightarrow{xy}$.

Step 2: Select some vertex x that is simple in H and initial with respect to U. If there is no such x then stop; H induces a subgraph with no simple vertex.
Step 3: Set $x < y$ for $Y \in H - x$, $H \leftarrow H - x$. If $H \neq \emptyset$, go to Step 1.
Step 4: Return the strong elimination order $<$.

This algorithm works because of the following fact: If H contains a simple vertex then H contains a simple vertex that is initial in U. Now since every induced subgraph of a strongly chordal graph contains a simple vertex, one will be found in Step 2. U is used to ensure that this simple elimination ordering is a strong elimination ordering.

We now present a characterization of coTT graphs based on an ordering property which we call a *proper ordering*.

Theorem 2.5 (Characterization I). A graph $G = (V, E)$ is coTT if and only if there is a total ordering $<$ of V so that whenever $xy \notin E$, either $x < N(y)$ or $y < N(x)$.

Proof. ($\Rightarrow$:) Let $G = (V, E)$ be a coTT graph with representation a_v and b_v for every $v \in V$. Obtain $<$ by ordering the vertices by nondecreasing b_v values. Suppose, in order to obtain a contradiction, that $xy \notin E$ and there are $w \in N(x)$ with $w < y$ and $z \in N(y)$ with $z < x$. Since $wx \in E$, $a_w \leq b_x$ and $a_x \leq b_w$; since $w < y, b_w \leq b_y$ by the choice of $<$. Similarly, $zy \in E$ implies that $a_z \leq b_y$ and $a_y \leq b_z$; since $z < x, b_z \leq b_x$. Together these imply that $a_x \leq b_y$ and $a_y \leq b_x$, which implies that $xy \in E$, a contradiction.

($\Leftarrow$:) Let $<$ be a total ordering of the vertices of $G = (V, E)$ satisfying the conditions of the theorem. Construct a coTT representation for G where b_v equals the position of vertex v in the ordering, and $a_v = \min\{b_w : w \in N(v)\}$. Note that $xy \in E$ implies that $a_x \leq b_y$ and $a_y \leq b_x$, by definition, and $xy \notin E$ implies that $a_x > b_y$ if $y < N(x)$, or $a_y > b_x$ if $x < N(y)$. ∎

To obtain the following corollary, we need only observe that every proper order is a strong elimination order.

Corollary 2.6. Every coTT graph is strongly chordal.

3. RECOGNITION ALGORITHM

Figure 2 illustrates the five *forbidden configurations* or *obstructions* that cannot occur as induced ordered subgraphs of a coTT graph; in each case $w < x < y < z$ in the ordering. In configurations (a), (c), and (d) the pair of vertices $yz \notin E$ violate the conditions of Theorem 2.5, and in configurations (b) and (e) the pair of vertices $xz \notin E$ violate Theorem 2.5. It is a simple task to check that these are the only forbidden configurations yielding the following theorem.

Theorem 3.1 (Characterization II). A graph $G = (V, E)$ is coTT if and only if there is a total ordering of the vertices with no obstruction of the form shown in Fig. 2. ∎

As we have previously noted, configurations (a) and (b) of Figure 2 are precisely those forbidden by strong elimination orderings. We introduce two rules that ensure configurations (c), (d), and (e) will never arise; conversely, the five forbidden configurations imply these two rules. Thus, proper orders are exactly strong elimination orders that obey these two rules.

Let $xywz$ be an induced P_4 in G, i.e., $xy, yw, wz \in E$ but $xw, xz, yz \notin E$. The first rule is that $x < z \langle=\rangle y < w$ in any proper ordering; we call this the *P_4 rule*. Let xy and wz induce a $2K_2$ in G, i.e., $xy, wz \in E$ but $xw, xz, yw, yz \notin E$. The second rule is that $x < w, \langle=\rangle x < z \langle=\rangle y < w \langle=\rangle y < z$ in any proper ordering we call this the *$2K_2$ rule*. Together we call these two rules the *PK rules*.

Our algorithm for determining if a graph is coTT or not proceeds as follows: First, Farber's Algorithm is used to ensure that the graph is strongly chordal. Next, we find a partial order on the vertices such that every linear extension satisfies the P_4 and $2K_2$ rules; we call such an order *conformist* since it always obeys these rules. We then show that this partial order can be extended to a strong elimination ordering using a modification of Farber's Algorithm. This ensures that a proper ordering is produced.

In order to simplify our discussion, we shall think in terms of orientations rather than orders. An order $<$ of a graph's vertices corresponds to an acyclic orientation U of the complete graph on the same vertex set (where $\vec{ab} \in U \leftrightarrow a < b$). Thus, to a given graph G we associate an order graph O_G, which is simply a complete graph on $V(G)$. Thus, we actually provide acyclic orientations of O_G. Orientations will be called *conformist, proper,* or *strong elimination* precisely if the corresponding orders are. We say that *x precedes y* (and *y follows x*) in an orientation U if $\vec{xy} \in U$. This formalism allows us to discuss "directed edges" rather than "ordered vertex pairs."

3.1. How to Conform

In a conformist orientation of O_G, the orientation of one edge of O_G may, through a sequence of applications of the P_4 and $2K_2$ rules, force the direction of many other edges. In fact, the edges of O_G can be partitioned into "forcing equivalence classes" such that the direction of one edge in a class determines the direction of every other edge in the class. More formally, we define a relation R on the edges of O_G such that $e_1 R e_2$ if the orientations of e_1 and e_2 are linked through a direct application of one of our two rules. Thus, the $2K_2$ rule yields the following:

(i) If ab, cd are a $2K_2$ then $acRad$, $acRbc$, $acRbd$, $adRbc$, $adRac$, $adRbd$, and $bcRbd$.

The P_4 rule gives

(ii) If $abcd$ is a P_4 then $abRbc$ and $bcRad$.

The transitive closure R^* of R is an equivalence relation on the edges of O_G. For any pair of vertices u and v we let $S(uv)$ be the equivalence class under R^* of the edge uv. Clearly, in any orientation obeying these rules, $S(uv)$ has one of two possible orientations: one containing $\vec{uv}$ and the other containing $\vec{vu}$. Note that these two orientations are mirror images so that one is acyclic if and only if the other is. It follows that if either of these two possible orientations is not acyclic then the graph is not coTT. We shall call an equivalence class *consistent* if this situation does not occur. The two possible orientations of a consistent equivalence class will also be called consistent.

The purpose of this subsection is to show that a conformist partial order can be obtained by orienting the non-singleton equivalence classes of a strongly chordal graph provided that all of the equivalence classes are consistent. The remainder of this subsection is devoted to proving the following theorem:

Theorem 3.1. Any strongly chordal graph all of whose equivalence classes are consistent has a conformist partial order.

We shall divide the edge-set of O_G into innocuous and dangerous edges. Call an edge uv *innocuous* if $S(uv)$ is a singleton. Call an edge uv *dangerous* if $S(uv)$ contains at least one other edge. If $S(uv)$ is a singleton then the two consistent orientations of this class are $\vec{uv}$ and $\vec{vu}$. It follows that in any acyclic orientation of O_G, every equivalence class consisting of an innocuous edge will have a consistent orientation. Thus, we need only concentrate on the dangerous edges of O_G since any acyclic orientation of the dangerous edges of O_G in which each non-singleton equivalence class has a consistent orientation will be conformist. So, we need only find such an orientation.

A naive way of doing so would be to arbitrarily choose one of the two consistent orientations on each non-singleton equivalence class and hope that the resulting orientation is acyclic. It turns out that any orientation constructed in this way must either be acyclic or contain a directed triangle. Furthermore, this directed triangle corresponds to one of two possible structures in the graph as described in the following lemma. These structures will be used in a decomposition approach to recursively generate a conformist orientation.

Lemma 3.1.1. Consider a strongly chordal graph $G = (V, E)$ all of whose equivalence classes are consistent. Arbitrarily choose one of the two orientations for each non-singleton equivalence class. One of two possible cases can occur.

(a) The resultant orientation is a conformist orientation.
(b) The resultant orientation U contains a directed triangle, i.e., vertices a, b, c where $\vec{ab}, \vec{bc}$, and $\vec{ca} \in U$ with one of the two possible induced subgraphs G, as shown in Figure 3.

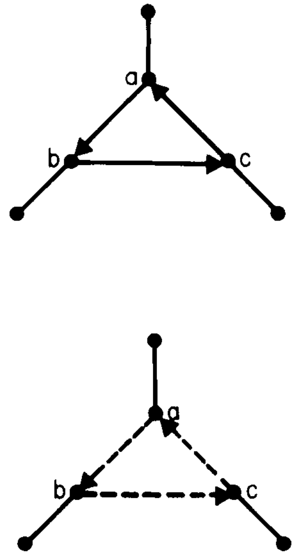

FIGURE 3. Cyclic triangles.

Proof. If the resultant orientation is a partial order, then we are done. If not, there must be a directed cycle of length at least three. [Since $S(xy) = S(yx)$, antisymmetry can only occur if it occurs within an equivalence class.] Let a, b, and c be three consecutive vertices on this cycle with $\vec{ab}$, $\vec{bc}$ in U. Since only non-singleton equivalence classes were directed, $S(ab)$ and $S(bc)$ must be non-singleton. It will be indicated how to show that $S(ac)$ is not a singleton, and that either $\vec{ac}$ yielding a shorter cycle or condition (b) holds. Repeating this argument proves the theorem.

We note that the P_4 and $2K_2$ rules imply that if $S(xy)$ is not a singleton, then there are vertices u and v such that $ux, vy \in E$ but $uy, vx \notin E$; we refer to this as the *PK rule*.

The proof proceeds by a case analysis on whether or not, a, b, and c are adjacent in G. We consider only the case where the three vertices induce a triangle. The other cases are proved similarly; the details can be found in [19].

Case 1. $ab, bc, ac \in E$.

Subcase (i). There is some x that $xb \in E$ and $xa, xc \notin E$. Since $S(ab)$ is not a singleton, the *PK* rule says there is a vertex y with $ya \in E$ and $yb \notin E$. Since G is chordal, $yx \notin E$. We can assume that $yc \notin E$, or else the P_4 rule applied to $yabx$ and $ycbx$ would imply that $\vec{cb} \in U$, a contradiction. Similarly, there is some z with $zc \in E$ and $zb, za, zx, zy \notin E$. Now the P_4 rule on $yacz$ implies that $S(ac)$ is not a singleton, and that either $\vec{ac} \in U$ or condition (b) holds.

Subcase (ii). There is no x such that $xb \in E$ and $xa, xc \notin E$. Since $S(bc)$ is not a singleton, by the PK rule there is some x that $xb \in E$ and $xc \notin E$. Also, since $S(ab)$ is not a singleton, by the PK rule there is some y such that $yb \in E$ and $ya \notin E$, and some z such that $za \in E$ and $zb \notin E$. Now xa and $yc \in E$; if not, we would be in Subcase (i). Since G is chordal, $xy, zy \notin E$. We may assume that $zc \notin E$; if not, $xz \notin E$ implies that $\{x, y, z, a, b, c\}$ induces a trampoline, and $xz \in E$ implies that $\{z, x, b, c\}$ induces a C_4. But now the P_4 rule applied to $zaby$ and $zacy$ implies that $\vec{ac}$. ∎

Case (a) of Lemma 3.1.1 yields the desired conformist orientation. The following lemma shows that the structures in Case (b) of Lemma 3.1.1 can be used to decompose the problem of finding a conformist orientation in G to one of finding a conformist orientation for two smaller induced subgraphs. So the problem can be solved recursively.

Lemma 3.1.2. Consider a strongly chordal graph $G = (V, E)$ all of whose equivalence classes are consistent. Let U be a consistent orientation of $O(G)$. Arbitrarily, choose one of the two possible orientations of every non-singleton equivalence class. If the orientation is cyclic then G can be partitioned into smaller subgraphs G_1 and G_2 so that a conformist orientation for G_1 and G_2 yields a conformist orientation for G.

From Lemma 3.1.1, we know that if the orientation is cyclic then G contains one of the two subgraphs shown in Fig. 3. The partition of G we will use depends on which of these two subgraphs is present. This is captured in the following four lemmas:

Lemma 3.1.2.1. Let G and U be as Lemma 3.1.2. Assume six vertices a, b, c, d, e, and f in G induce $3K_2'$s as depicted in Figure 3 with $\vec{ab}$, $\vec{bc}$, and $\vec{ca}$ in U. Set $A = \{x \mid x \text{ sees all of } a, b, c, d, e, f\}$. Then

(i) A is a clique.
(ii) The $3K_2'$s are in distinct components of $G - A$. Furthermore, every vertex in A sees all the vertices in these three components.

Proof. Since G is chordal, A is a clique. The proof of Property (ii) depends upon repeated application of the following fact:

Fact 1. If G contains a $3K_2$, as in Figure 3, with $\vec{ab}$, $\vec{bc}$, and $\vec{ca}$ in U, then no vertex of G sees a vertex in two of the three K_2's and misses a vertex from the third.

Proof. Assume x sees a and c, but misses b. Then $bexa$ and $bexc$ are both either P_4's (if e sees x) or $2K_2'$s (if e misses x). In either case, $\vec{ab}$ and $\vec{cb}$ are in the same equivalence class. This contradicts the fact that $\{a, b, c\}$ is a directed triangle in U. By symmetry, we established Fact 1. ∎

It follows from Fact 1 that every vertex of $G - \{a, b, c, d, e, f\}$ either sees all of $\{a, b, c, d, e, f\}$ or sees vertices from at most one of the three $K_2's$. Let $A = \{y \mid y$ sees all of $a, b, c, d, e, f\}$. We want to show that no two of the $K_2's$ are in the same component of $G - A$. This will follow easily from the following observation: Assume x in $G - A$ sees a vertex from one of the $K_2's$. Replace the other vertex of this K_2 by x. We obtain a new $3K_2$ in G with analogous orientation. Also, from Fact 1, we see that a vertex sees all of the original six vertices if it sees all six of the vertices in the new $3K_2$. Now, assume that, for any subgraph of G as in Figure 3 we have two of the $2K_2's$ in the same component of $G - A$. Then choose the shortest path in $G - A$ between vertices in distinct $K_2's$. Furthermore, choose the minimum such path over all appropriate choices of subgraphs and corresponding sets A. Denote the two endpoints of this path by u and v. Let x be the first vertex on this u to v path. Recall that replacing u by x gives a new $3K_2$. Furthermore, the set A corresponding to this new $3K_2$ is the same as that for the old $3K_2$. Now, we have a path in $G - A$ from x to v. This path contradicts the minimality of the path from u to v. Thus, for any subgraph, as in Figure 3, with $\{a, b, c\}$ forming a triangle in U, each K_2 is in a separate component of $G - A$. Call these components C_1, C_2, and C_3. Recall that if $x \in G - A$ is adjacent to any vertex in a K_2 then we saw that x must see all of A. By induction on the length of paths, it follows that each vertex in A sees all of $C_1 \cup C_2 \cup C_3$. ∎

Lemma 3.1.2.2. Let G be a strongly chordal graph partitioned into sets A, C_1, C_2, C_3, and N as in Lemma 3.1.2.1. Assume we have a conformist orientation U_1 of the dangerous edges of O_{G-C_3} and a conformist orientation U_2 of the dangerous edges of O_{C_3}. Then we can obtain a conformist orientation U_3 of the dangerous edges of O_G by the following:

(i) Orienting the dangerous edges of O_{G-C_3} as in U_1.
(ii) Orienting the dangerous edges of O_{C_3} as in U_2.
(iii) Let c be an arbitrary vertex of C_1. For x in C_3 and y in $G - C_3 - C_1$, $\vec{xy}$ ($\vec{yx}$) is in U_3 if $\vec{cy}$ ($\vec{yc}$) is in U_1.
(iv) For x in C_3 and y in C_1, $\vec{xy}$ is in U_3.

Proof. If xy is a dangerous edge of O_{G-C_3} or O_{C_3} then xy is still dangerous in O_G. The only new dangerous edges are those created by $2K_2's$ or $P_4's$, which are partially but not entirely in C_3. It is a tedious but routine matter to verify that any such P_4 or $2K_2$ is of one of the five types listed below.

(1) $2K_2$ x_1x_2, n_1n_2 with $x_1, x_2 \in C_3$ and $n_1, n_2 \in N$.
(2) $2K_2$ x_1x_2, c_1c_2 with $x_1, x_2 \in C_3$ and $c_1, c_2 \in C_1$.
(3) $2K_2$ x_1x_2, c_1c_2 with $x_1, x_2 \in C_3$ and $c_1, c_2 \in C_2$.
(4) $2K_2$ x_1a_1, n_1n_2 with $x_1 \in C_3$, $a_1 \in A$ and $n_1, n_2 \in N$.
(5) P_4 $x_1a_1n_1n_2$ with $x_1 \in C_3$, $a_1 \in A$ and $n_1, n_2 \in N$.

Given a $2K_2$ of type 4 or a P_4 of type 5 we can replace x_1 by c to obtain a corresponding $2K_2$ or P_4 in $G - C_3$. It follows that if xy is a dangerous edge arising from such a $2K_2$ or P_4 (with $x \in C_3$) then cy is a dangerous edge of O_{G-C_3}. Furthermore, the orientation of vertices of $2K_2's$ and $P_4's$ of types 4 and 5 in U_3 obey the P_4 and $2K_2$ rules because the corresponding $2K_2's$ and $P_4's$ in U_1 do.

Similarily, given a $2K_2$ of Type 1 or 3, we can replace the $K_2 x_1 x_2$ by the K_2 cc', where c' is some neighbor of c in C_1. Thus, if xy is a dangerous edge arising from a $2K_2$ of Type 1 or 3 (with x in C_3) then cy is a dangerous edge of O_{G-C_3}. Furthermore, these kinds of $2K_2's$ obey the $2K_2$ rule because the corresponding $2K_2's$ obey the $2K_2$ rule in U_1.

The $2K_2's$ of Type 2 are the only ones that give dangerous edges of the form xy with $x \in C_3$, $y \in C_1$. Since C_1 and C_3 are both connected, it follows that every such edge of O_G will be dangerous. Furthermore, it is clear that our orientation of these edges will obey the $2K_2$ and P_4 rules.

Thus, we can construct our orientation U_3 as described in the lemma and it will satisfy the $2K_2$ and P_4 rules. We need only ensure that U_3 is also acyclic. Let $C = \{v_0, v_1, \ldots, v_{k-1}\}$ be a cycle in O_G. Furthermore, choose C to have the minimal number of vertices in O_G over all such cycles. And further, choose C to have the least number of vertices of C_3 subject to the previous conditions. C must contain vertices v_{i-1} and v_i (all addition is modulo k) such that $v_i \in C_3$ and $v_{i-1} \in G - C_3$; otherwise, C would be a cycle in U_1 or U_2. If v_{i+1} is not in $C_1 \cup C_3$ then we can replace v_i by c, our special vertex of C_1, to obtain a cycle $C - v_i + c$ in O_G; this would contradict the minimality of C. If v_{i+1} is in C_3 then $v_{i-1}v_{i+1}$ is an edge of U_3, and thus $C - v_i$ is a cycle in O_G; again, we obtain a contradiction. Therefore, v_{i+1} is in C_1. Now, the edge $v_{i-1}v_i$ arises from a $2K_2$ or P_4 of Type 1, 3, 4, or 5. As before, we can find a corresponding P_4 or $2K_2$ using v_{i+1} (rather than c). Since U_3 obeys the $2K_2$ and P_4 rules, clearly $\overline{v_{i-1}v_{i+1}}$ is an edge of U_3. Thus, $C - v_i$ is a cycle in O_G, contradicting the minimality of C. This demonstrates that U_3 is acyclic, and therefore a conformist orientation of O_G as required. ∎

Lemma 3.1.2.3. Assume we have vertices a, b, c, d, e, and f in G that induce the subgraph depicted in Figure 3 with $\vec{ab}$, $\vec{bc}$, and $\vec{ca}$ in U. Then we can partition G into three stable sets S_1, S_2, and S_3; three cliques C_1, C_2, and C_3; and sets A, B, and N such that

(i) $x \in A \Rightarrow x$ sees all of $S_1 \cup S_2 \cup S_3 \cup C_1 \cup C_2 \cup C_3$.
(ii) $x \in N \Rightarrow x$ misses all of $S_1 \cup S_2 \cup S_3 \cup C_1 \cup C_2 \cup C_3$.
(iii) $x \in B \Rightarrow x$ sees all of $C_1 \cup C_2 \cup C_3$ and misses all of $S_1 \cup S_2 \cup S_3$.
(iv) S_i misses C_j for $i \neq j$.
(v) Each vertex of S_i sees a vertex of C_i and vice versa for $i = 1, 2, 3$.

Proof. The proof of this lemma is similar to that of Lemma 3.1.2.1 and can be found in [19]. ∎

Lemma 3.1.2.4. Let G be a strongly chordal graph partitioned into sets C_1, C_2, C_3, S_1, S_2, S_3, A, B, and N as in Lemma 3.1.2.3. Assume we have conformist orientations U_1 and U_2 of the dangerous edges of $O_{G-C_3-S_3}$ and $O_{S_3 \cup C_3}$, respectively. Then we can obtain a conformist orientation U_3 of the dangerous edges of O_G as follows:

(i) Orient the dangerous edges of $O_{G-C_3-S_3}$ as in U_1.
(ii) Orient the dangerous edges of $O_{C_3 \cup S_3}$ as in U_2.
(iii) Let c be an arbitrary vertex of C_1. For x in C_3 and y in $G - C_3 - S_3 - C_1 - S_1$, $\vec{xy}$ ($\vec{yx}$) is in U_3 if and only if $\vec{cy}$($\vec{yc}$) is in U_1.
(iv) For x in S_3 and y in S_1, $\vec{xy}$ is in U_3. For x in C_3 and y in C_1, $\vec{xy}$ is in U_3.

Proof. The proof of this lemma is similar to that of 3.1.2.2 and can be found in [19]. ∎

3.2. How to Be Proper

The purpose of this subsection is to show that a conformist partial order for a strongly chordal graph can be extended to a strong elimination ordering. Together these results imply that the resultant order is a proper ordering. This is proved in the following theorem by an extension of Algorithm 1.

Lemma 3.2. Consider a strongly chordal graph $G = (V, E)$ all of whose equivalence classes are consistent. Let P be a conformist orientation produced by Lemma 3.1.2. P can be extended to a proper ordering $<$ for G.

Proof. Let G be a strongly chordal graph all of whose equivalence classes are consistent. Using the procedures outlined in Lemma 3.1.2 we can construct a conformist orientation U of the dangerous edges of $O(G)$. The following modified version of Farber's algorithm will give us a strong elimination ordering that extends U.

Algorithm 1′

Input: A strongly chordal graph G and a conformist orientation U of the dangerous edges of $O(G)$.
Output: A strong elimination order on G such that if $\vec{ab} \in U$ then $a < b$.

At any given time, we will have a subgraph H of vertices of G whose position in the order has not been determined. We know that $G - H$ precedes H and $<$ is a total order on $G - H$.

Step 0. Set $H \leftarrow G$, $< \leftarrow \varnothing$.
Step 1. For each pair of adjacent vertices x and y in H, if $N_H(x) \subset N_H(y)$ then set $U \leftarrow U + \{\vec{xy}\}$.
Step 2. Select some x that is simple in H and initial with respect to U in H.
Step 3. Set $x < y$ for y in $H - x$, set $H \leftarrow H - x$. If H is non-empty then go to Step 1.
Step 4. Return $<$.

We will show that this algorithm works in two steps. First, we will show that at all times U remains acyclic and anti-symmetric. We will then show that H always contains a simple vertex that is initial with respect to U.

When we initialize the algorithm, we know that U is acyclic and anti-symmetric. If U becomes symmetric at any time, it must be through an application of Step 1. Assume $\vec{ab}$ is added in Step 1; we will show that $\vec{ba}$ is not in U. Clearly $\vec{ba}$ cannot have been added to U in Step 1 because than $N_H(b) \subset N_H(a)$, contradicting $N_H(a) \subset N_H(b)$. Thus, $\vec{ba}$ must be the mid-edge of some P_4 $xbay$ in G. Now, when we added $\vec{ab}$ to U, we knew there was a vertex z in H that saw b but not a. Since G is chordal z misses y; thus $zbay$ is a P_4. Since $\vec{ba}$ is U, by the P_4 rule, so is $\vec{zy}$. Since z is in H, so is y. But now y sees a but not b, contradicting $N_H(a) \subset N_H(b)$. The preceding remarks show that at all times U is anti-symmetric.

To see that U also remains acyclic, we will need a few more facts about directed paths in U. We shall call arcs of U added in Step 1 *Farber edges*. Those arcs of U with which the algorithm was initiated shall be refereed to as *PK* edges.

Fact 5. Consider vertices a, b, and c of H. If $\vec{ab}$ and $\vec{bc}$ are Farber edges then so is $\vec{bc}$.

Proof. Note first that ac is an edge of G since $N_H(b) \subset N_H(c)$. Assume $\vec{ab}$ was added to U after $\vec{bc}$. At the instant $\vec{ab}$ was added we have $N_H(a) \subseteq N_H(b) \subset N_H(c)$. Thus, $\vec{ac}$ must also be a Farber edge of U. Similarily, if $\vec{bc}$ was the second edge to be added, then when it was added we had $N_H(a) \subset N_H(b) \subset N_H(c)$. Thus, in either case, $\vec{ac}$ is a Farber edge of U. ▮

Fact 6. Consider vertices a, b, and c of H. If $\vec{ab}$ and $\vec{bc}$ are *PK* edges of $O(G)$, then so is $\vec{ac}$.

Proof. This follows from our proof of Lemma 3.1.1. ▮

Fact 7. Consider vertices a, b, and c of H. If $\vec{ab}$ is a Farber edge, $\vec{bc}$ is a *PK* edge and a sees c, then $\vec{ac}$ is an edge of U.

Proof. Since $\vec{ab}$ is a Farber edge, $N_H(a) \subset N_H(b)$ and so b sees c. If $\vec{bc}$ is a Farber edge, then by Fact 5 we would know that $\vec{ac}$ is a Farber edge. Furthermore, $\vec{cb}$ is not a Farber edge because we have already shown that U remains anti-symmetric. Thus, we can assume that bc is not a Farber edge in either direction. Since $\vec{bc}$ is a *PK* edge, there is a P_4 $xbcy$ in G. Thus, $N_G(b)$ is incomparable with $N_G(c)$. If at the instant $\vec{ab}$ is added to $UN_H(c) \subseteq N_H(b)$, then bc would have to have been made a Farber edge in one of the two possible directions. It follows that some d in H sees c but not b. Now d misses a since $N_H(a) \subset N_H(b)$. Furthermore, a sees some e in H, which misses c; otherwise, $\vec{ac}$ would be a Farber edge. Since $N_H(a) \subset N_H(b)$, b sees e. Furthermore, since

G is chordal, e misses d. Now $ebcd$ and $eacd$ are both $P_4's$ of G, and by repeated application of the PK rule, we see $\vec{bc}$ is a PK edge of U. ∎

Fact 8. Consider vertices a, b, and c of H. If $\vec{ab}$ is a Farber edge, $\vec{bc}$ is a PK edge, and c misses both a and b, then $\vec{ac}$ is an edge of U.

Proof. By the PK rule, some x in G sees c but not b. Now $baxc$ induces either a $2K_2$ or a P_4. In either case, $\vec{ax}$ is in U, and since a is in H, so is x. But now, since $N_H(a) \subset N_H(b)$, x misses a. It follows that ab, xc is a $2K_2$, and $\vec{ac}$ is in U. ∎

Fact 9. Consider vertices a, b, and c of H. Assume $\vec{ab}$ is a PK edge of U, $\vec{bc}$ is a Farber edge of U, and a sees b, then $\vec{ac}$ is an edge of U.

Proof. Since $\vec{bc}$ is a Farber edge, $N_H(b) \subset N_H(c)$; thus, a sees c. Since ab is a PK edge, there is a P_4 $xaby$ in G. As ab is not a Farber edge in either direction, there is some d in H, which sees a but not b. Since G is chordal, d misses y. Now, $daby$ is a P_4 so $\vec{dy}$ is in U. Since d is in H, so is y. If $N_H(a) \subset N_H(b)$, then $\vec{ac}$ would be a Farber edge. Otherwise, some e in H sees a but not c. Since G is chordal, e misses y. Since $N_H(b) \subset N_H(c)$, e misses b. Now $eaby$ and $eacy$ are $P_4's$ and it follows that $\vec{ac}$ is in U. Consider now a shortest path $P = \{x = p_0, p_2, \ldots, p_n = y\}$ in U between any two vertices x and y of H. Facts 5 and 6 imply that this path must alternate between PK and Farber edges. Facts 7 and 8 imply that a PK edge that follows a Farber edge must correspond to a nonedge of G. However, Fact 9 implies that a PK edge that precedes a Farber edge must correspond to an edge of G. Thus, we cannot have a Farber, PK, Farber sequence of edges on P. It follows that any shortest path in U must have at most three edges. Furthermore, consider such a path $P = \{p_0, p_1, p_2, p_3\}$ with three edges. We know that $\overline{p_0 p_1}$ and $\overline{p_2 p_3}$ are PK edges while $p_1 p_2$ is a Farber edge. Also, Facts 7, 8, and 9 imply that p_0 misses p_1 and p_2 in G, while p_3 sees p_2 but misses p_1. Now, if p_0 saw p_3 then $p_0 p_3 p_2 p_1$ would be a P_4. But then $\overline{p_3 p_2}$ would be in U, contradicting the fact that U is anti-symmetric. Thus, p_0 misses p_3. Now, by the PK rule, $p_0 z p_1 p_2$ would be a $2K_2$ and $p_0 p_2$ would be in U, contradicting the minimality of P. Thus, z sees p_2. Now, $p_0 z p_2 p_1$ is a P_4 so $\overline{z p_2}$ is in U. Clearly, z sees p_3, as otherwise $p_0 z p_2 p_3$ would be P_4 and, by the P_4 rule, $p_0 p_3$ would be in U. Now, $z p_2$ is a PK edge of G and $\overline{p_2 p_3}$ is a PK edge of G, so by Fact 6, $\overline{z p_3}$ is a PK edge of G. If $\overline{p_0 z}$ were a Farber edge in U, then $p_0 z p_3$ would be a shorter path between p_0 and p_3. Thus, there is some a in H that sees p_0 but not z. Now, since G is chordal, a misses $\{v_1, v_2, v_3\}$. But then ap_0, $p_1 p_2$ and ap_0, $p_2 p_3$ are $2K_2's$. It follows that $\overline{p_0 p_3}$ is in U, contradicting the minimality of P. Thus, the shortest path in U between any two vertices has length at most two.

We turn now to the shortest cycle in U. By the above remark, this must be a triangle. But this triangle must have either two PK edges or two Farber edges. In the first case, by Fact 6, we will contradict the fact that U is anti-symmetric.

In the second case, Fact 5 leads to the same contradiction. Thus, U contains no triangles. It follows that U remains acyclic throughout our application of Algorithm 1′. ∎

We will now show that after each application of Step 1 of Algorithm 1′, H contains a simple vertex that is initial with respect to U. Since G is strongly chordal, H contains a simple vertex. Since U is acyclic, H contains a simple vertex x such that $E = \{y \mid \text{there is a } y \text{ to } x \text{ path in } U\}$ contains no vertex that is simple in H. We will assume that $E = \emptyset$ and obtain a contradiction.

Fact 10. No y in E sees x.

Proof. Recall that the shortest path from y to x has at most two edges. Facts 5–9 imply that if xy is an edge of G and there is a path of length two from y to x in G, then $\vec{xy}$ is an edge in U. Farber showed that if $\vec{yx}$ is a Farber edge and x is simple in H, then y is simple in H, contradicting our choice of x. Consider now a y in E that is adjacent to x. We know $\vec{xy}$ is a PK edge of U and not a Farber edge of U. Now, in G there is a P_4 $axyb$. Thus, $N_G(x)$ is incomparable to $N_G(y)$. Since xy is not a Farber edge in either direction, we know that some c in H sees y but not x. Since G is chordal, c misses a. Now $cyxa$ is a P_4 and since $\vec{yx}$ is in U, so is $\vec{ca}$. This implies that since c is in H, so is a. But now x sees y and a in H which are non-adjacent. This contradicts the fact that x is simple in H. The desired result follows. ∎

Fact 11. yx is a dangerous edge of $O(G)$ for every y in E.

Proof. Consider a shortest path in U from y in E to x. We know that this path has at most two edges. Assume the path has two edges, then there is a z in E, such that the path is yzx. Now, by Fact 10, z misses x and is a PK edge of U. If $\vec{yz}$ is a PK edge of U then, by Fact 6, $\vec{yx}$ is an edge of U. If $\vec{yz}$ is a Farber edge of U then, by Fact 8, $\vec{yx}$ is an edge of U. In either case, we contradict the minimality of our path. Thus, for every y in E, $\vec{yx}$ is in U. However, by Fact 10, y misses x in G; so this must be a PK edge, implying Fact 11. ∎

Now, set $A = \{y \mid y \in H - E,\ y \text{ misses } x, y \text{ sees some } z \text{ in } E\}$
$B = \{y \mid y \in H - E - A,\ y \text{ misses } x, y \text{ sees some } z \text{ in } A\}$
$C = \{y \mid y \in H - E - A - B,\ y \text{ misses } x\}$.
Then $H = E + A + B + C + x + N(x)$. Farber has shown that any strongly chordal graph is either a clique or contains two non-adjacent simple vertices. Since E is non-empty, and x misses every element of E, it follows that $H - C$ is not a clique. Thus, $H - C$ contains two non-adjacent simple vertices. We shall now show that one of these vertices is in E, and that this vertex is simple in H. We note first that $x + N_H(x)$ is a clique. Thus, $H - C - x - N_H(x)$ contains at least one vertex that is simple in $H - C$. Note that $H - C - x - N_H(x) = E + A + B$. Consider a vertex a in A. By definition, a sees some h in H. By Fact 11, hx is a dangerous edge of $O(G)$, so h misses some y in $N_G(x)$.

Now, if a missed y, then ha, xy would be a $2K_2$ and $\vec{ax}$ would be in U. But a is in A not E and so a must see Y. Now, $hayx$ is a P_4 and so $\vec{ay}$ is in U. Since a is in H, this implies that y is in H. Thus, a sees y and h, which are non-adjacent. If follows that no vertex of a is simple in $H - C$. Consider a vertex b in B. We know that b sees some a in A. As above, a sees h in H and y in $N_H(x)$ such that y misses h. Now $a, y \in N_{H-C}(b)$ but a and y have incomparable neighborhoods in $H - C$ (since x sees y but not a, and h sees a but not y). Thus, no vertex of B is simple in $H - C$.

To summarize, we know some vertex of $H - C - x - N_H(x) = E + A + B$ is simple in $H - C$. Furthermore, we know that no vertex of $A \cup B$ is simple in $H - C$. Thus, some vertex y in E must be simple in $H - C$. We claim that y is simple in H. Clearly no vertex in C sees y. Thus, y is still simplicial in it. If y were not simple we would have vertices $\{c, e, f, g\}$ of H such that y saw e and f, e saw c but not g, and f saw s but not c. Since e is simple in $H - C$, we can assume that $c \in C$. Now, c misses $E \cup A + x$. Also, y misses $B \cup C + x$. Thus, e must be in $N_H(x)$. If f were in $N_H(x)$, we could replace y by x, contradicting the fact that x is simple in H. ∎

This completes the proof that Algorithm 1′ extends a conformist partial order for a strongly chordal graph to a strong elimination order.

3.3 An End to Propriety

We note that the results of sections 3.1 and 3.2 give two additional characterizations of coTT graphs, one of which yields a polynomial-time recognition algorithm. Define a graph to be a *PK graph* if there is a total ordering $<$ on the vertices that satisfies the P_4 and $2K_2$ rules.

Theorem 3.3.1 (Characterization III). A graph G is a coTT graph if and only if G is a strongly chordal graph and a PK graph. ∎

Theorem 3.3.2 (Characterization IV). A strongly chordal graph is coTT if and only if each PK equivalence class is consistent. ∎

The verification of Theorem 3.3.2 also yields a polynomial-time recognition algorithm for coTT graphs.

Algorithm 2

Input: A graph $G = (V, E)$.
Output: A proper order if G is coTT, or either an induced subgraph with no simple vertex, or a cyclic PK equivalence class, otherwise.

Step 1. Check to see if G is strongly chordal by applying Algorithm 1. If G is not strongly chordal then stop; G is not coTT.

Step 2. Apply the P_4 and $2K_2$ rules to form the equivalence classes. If any equivalence class is not consistent then stop; G is not coTT.

Step 3. Arbitrarily choose one of the two orientations for each non-singleton equivalence class. If the orientation is conformist then go to Step 4; if not, partition the graph into smaller subgraphs and apply the algorithm recursively to form a conformist orientation for G as in the proof of Theorem 3.1.

Step 4. Use Algorithm 1′ to extend the conformist orientation to a proper order.

We note that Algorithm 2 provides a proper order for a coTT graph G. From this order, we can obtain weights and tolerances for each vertex that satisfy the requirement for a threshold tolerance representation for G using the proof of Theorem 2.5. If we only want to check if G is coTT, we need only use Steps 1 and 2 of Algorithm 2 to check that G is strongly chordal and form the P_4 and $2K_2$ equivalence classes, and to check to see if they are consistent. Step 3 can be thought of as constructing a binary decomposition tree with G as the root. Each time we split a graph G we make two children G_1 and G_2, as described in Lemma 3.1.2. The leaves of the tree are disjoint subgraphs and so we apply Algorithm 2 at most $2 \cdot |V|$ times. It should be clear that since this partitioning can be done in polynomial-time, so can the entire Algorithm 2.

Theorem 3.3.3. The algorithm correctly recognizes coTT graphs.

Proof. By Corollary 2.6, a coTT graph is strongly chordal, so if G is found not to be strongly chordal in Step 1 then G is not coTT. Every proper ordering must be consistent with the equivalence classes, so if some equivalence class itself is not consistent then no ordering exists in Step 2. Theorem 3.1 ensures that a conformist orientation is found in Step 3. Theorem 3.2 ensures that a proper order is found in Step 4. ∎

4. CONCLUDING REMARKS

We have introduced a class of graphs generalizing threshold graphs by adding threshold tolerances. We have obtained several characterizations of these graphs and a polynomial-time recognition algorithm. We have also shown that the complements of these graphs contain the class of interval graphs and are contained in both the classes of strongly chordal graphs and interval tolerance graphs.

Benzaken et al. [1] also studied a generalization of threshold graphs, which they called threshold signed graphs. These graphs are incomparable to coTT graphs since C_4 is in their class but not ours, and the graph in Fig. 1(a) is in our class but not theirs.

Chordal graphs [3, 10, 23] and strongly chordal graphs [7] are also characterized in terms of intersection graphs of certain subtrees in a tree. These and

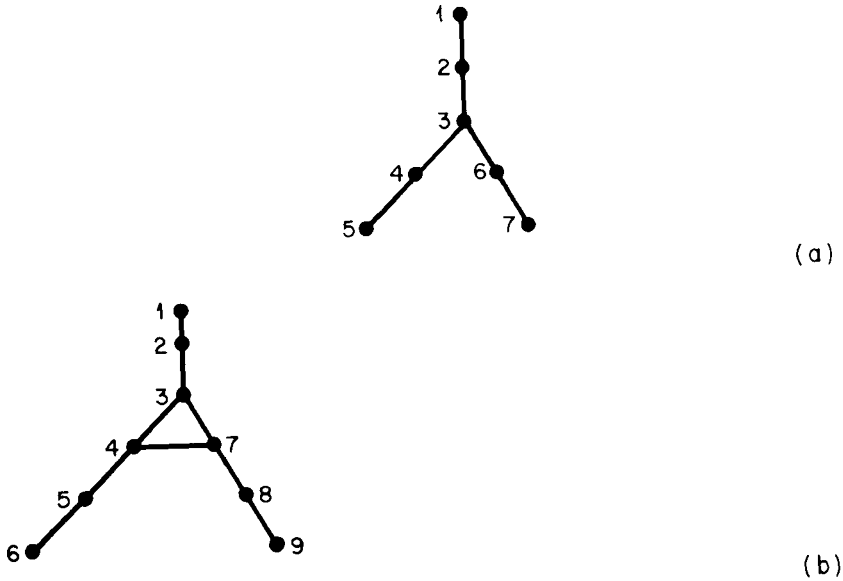

FIGURE 4. Forbidden subgraphs for coTT graphs.

other classes of graphs arising as the intersection graphs of paths in a tree are studied in [20]. We leave such a characterization for coTT graphs as an open problem.

Another open problem is to characterize coTT graphs in terms of forbidden induced subgraphs. A partial list of forbidden subgraphs is given in Fig. 4. [4] characteristic threshold graphs as those graphs with no induced C_4, P_4, or $2K_2$. We also leave as an open question the characterization of PK graphs.

ACKNOWLEDGMENTS

We would like to thank Michael E. Saks of Bellcore for pointing out Theorem 2.5 to us.

References

[1] C. Benzaken, P. L. Hammer, and D. deWerra, Threshold characterization of graphs with Dilworth number two. *J. Graph Theory*, to appear.

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

[3] P. Buneman, A characterization of rigid circuit graphs. *Discrete Math.* **9** (1974) 205–212.

[4] V. Chvatal and P. L. Hammer, Aggregation of inequalities in integer programming. *Ann. Discrete Math.* **1** (1977) 145–162.

[5] G. J. Chang and G. L. Nemhauser, The k-domination and k-stability problem on sun-free chordal graphs. *SIAM J. Algebraic Discrete Methods* **5** (1984) 332–345.

[6] G. A. Dirac, On rigid circuit graphs. *Abh. Math. Sem. Univ. Hamburg,* **25** (1961) 71–76.

[7] M. Farber, Applications of LP duality to problems involving independence and domination. Ph.D. thesis, Rutgers University, 1982.

[8] M. Farber, Characterizations of strongly chordal graphs. *Discrete Math.* **43** (1983) 173–189.

[9] D. R. Fulkerson and O. A. Gross, Incidence matrices and interval graphs. *Pac. J. Math.* **15** (1965) 835–855.

[10] F. Gavril, The intersection graph of subtrees in trees are exactly the chordal graphs. *J. Combinat. Theory B* **16** (1974) 47–56.

[11] P. C. Gilmore and A. J. Hoffman, A characterization of comparability graphs and interval graphs. *Can. J. Math.* **16** (1964) 539–548.

[12] M. C. Golumbic, Threshold graphs and synchronizing parallel processes. *Combinatorics, Colloquium of the Mathematics Society Janos Bolyai,* Vol. 18 (1978) 419–428.

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

[14] M. C. Golumbic and C. L. Monma, A generalization of interval graphs with tolerances. *Proceedings of the 13th Southeastern Conference on Combinatorics, Graph Theory and Computing, Congressus Numeration,* Vol. 35 (1982), Utilatis Mathematics, Winnepeg, 321–331.

[15] M. C. Golumbic, C. L. Monma, and W. T. Trotter, Jr., Tolerance graphs. *Discrete Appl. Math.* **9** (1984) 157–170.

[16] A. Hajnal and J. Surányi, Uber die auflösung von Graphen in vollständige Teilgraphen. *Ann. Univ. Sci. Budapest Eötvös Sect. Math.* **1** (1958) 113–121.

[17] P. B. Henderson and Y. Zalcstein, A graph theoretic characterization of the PV chunk class of synchronizing primitives. *SIAM J. Comput.* **6** (1977) 88–108.

[18] C. G. Lekkerkerker and J. C. Boland, Representation of a finite graph by a set of intervals on the real line. *Fund. Math.* **51** (1962) 45–64.

[19] C. L. Monma, B. A. Reed, and W. T. Trotter, Jr., Threshold tolerance graphs. Bellcore technical report (1986).

[20] C. L. Monma and V. Wei, Intersection graphs of paths in a tree. *J. Combinatorial Theory B,* **41** (1986) 141–181.

[21] J. Orlin, The minimal integral separation of a threshold graph. *Ann. Discrete Math.* **1** (1977) 415–419.

[22] D. J. Rose, Triangulated graphs and the elimination process. *J. Math. Anal.,* **32** (1970) 597–609.

[23] J. R. Walters, Representatives of chordal graphs as subtrees of trees. *J. Graph Theory* **2** (1978) 265–267.