Journal Title: Congressus Numerantium

Volume: 55
Issue:
Month/Year: 1986
Pages: 187 - 197

Article Author: Monma, Reed and Trotter

Article Title: A generalization of threshold graphs with tolerance

os, N. :"Partition Seman-
the 4th ACM Symposium on
land, Oregon, 1985, pp.

bras and Their Applica-
n and Control, vol. 7,

ses, Computer Science

A Generalization of Threshold Graphs with Tolerances

Clyde L. Monma
Bell Communications Research
Morristown, New Jersey 07960

Bruce Reed
McGill University
Montreal, Canada

William T. Trotter, Jr.
University of South Carolina
Columbia, South Carolina

ABSTRACT

In this extended abstract, we introduce a class of graphs which 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. A final paper complete with all proofs will appear at a later time.

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

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

If we insist that all tolerance be equal, we obtain the class of threshold graphs [CH77]; see also [Go78; Go80, Chapter 10; HZ77; Or77]. It is easy to see that we may require that all weights and tolerances are positive, and that strict inequality holds in (*).

For our purposes, it is convenient to present our results in terms of the complement of threshold tolerance graphs, which we call coTT graphs. An equivalent definition is that 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

$$xy \in E <=> a_x \leq b_y \text{ and } a_y \leq b_x.$$

To see that these definitions are equivalent, 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 [BL76; FG65; GH64; Go80, Chapter 8; LB62] 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,

$$xy \in E <=> I_x \cap I_y \neq \emptyset.$$

A graph $G = (V, E)$ is called an interval tolerance graph [GM82, GMT84] if there are intervals $I_v = [L_v, R_v]$ and tolerances τ_v for each $v \in V$ so that

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

where $|I|$ is the length of interval I. A graph $G = (V, E)$ is called a chordal graph [Bu74; Di64; FG65; Ga74; HS58; LB74; Ro70; Wa78] 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. □

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 [Fa83] (also called sun-free [CN84] graphs). In Section 3, we present alternate characterizations of coTT graphs, one of which leads to a polynomial-time algorithm for recognizing coTT graphs.

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∈E; otherwise we say that x misses y. An independent set is a set of vertices with each pairs missing each other. A clique is a set of vertices with each pairs seeing each 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 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 are compatible in the graph G if $N(x) \subseteq N(y)$ 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 [Fa83] if every induced subgraph has a simple vertex. A similar characterization holds for chordal graphs.

Theorem 2.1. [Di61,LB62]

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 [Fa83] obtains a forbidden subgraph characteristic for strongly chordal graphs. A trampoline is a graph $G = (V, E)$ on $2n$ vertices for some $n \geq 3$ whose vertices can be partitioned into $W = \{w_1, w_2, \cdots, w_n\}$ and $U = \{u_1, u_2, \cdots, u_n\}$ so that W is independent, U forms a clique, and w_i is adjacent to u_j if and only if $i = j$ or $i \equiv j + 1 \pmod{n}$. Figure 1(b) is a trampoline with n=3.

Theorem 2.2. [Fa83]

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 orders. 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 which has no other vertex preceding if 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 [Ro70] 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 [Fa83] 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 [Fa83] 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 2(b) occur. (The order is given by w < x < y < z.) We note that elimination orders forbid exactly Figure 2(a).

Theorem 2.3. [FG65, Ro70]

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. [Fa83]

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. □

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

Theorem 2.5. (Characterization I)

A graph $G = (V, E)$ is coTT if and only if there is an ordering $<$ on V so that whenever $xy \notin E$, either $x < N(y)$ or $y < N(x)$.

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 which can not 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 an ordering of the vertices with no obstruction of the form shown in Figure 2. □

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

Let $xywz$ be an induced P_4 in G, i.e., xy, yw, $wz \in E$ but xw, $yz \notin E$. The first rule is that $x < z \iff y < w$ in any proper order; 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 $x < w \iff x < z \iff y < w \iff w < z$ in any proper order; we call this the $2K_2$ rule.

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 all rules. We then show that this partial order can be extended to a strong elimination order using a modification of Farber's algorithm. This ensures that a proper order 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 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

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 3.1.1.

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

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:

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

While the P_4 rule gives:

(ii) If abcd is a P_4 then ad R bc.

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 $\vec{uv}$, 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.

We shall now 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 large (i.e. 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 3.1.2.

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 order is a partial order on the vertices.

(b) The resultant order contains a directed triangle, i.e., vertices a, b c with $a < b < c < a$, with one of the two possible included subgraphs G as shown in Figure 3.

Case (a) of Lemma 3.1.2 yields the desired conformist order. The following lemma shows that the structures in Case (b) of Lemma 3.1.2 can be used to decompose the problem of finding a comformist order in G to one of finding a conformist order for two smaller induced subgraphs. So the problem can be solved recursively.

Lemma 3.1.3.

Consider a strongly chordal graph G=(V,E) all of whose equivalence classes are consistent. 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 order for G_1 and G_2 yields a conformist order for G.

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 order. Together these results imply that the resultant order is a proper order. This is proved in the following theorem by an extension of Farber's algorithm.

Theorem 3.2.1

Consider a strongly chordal graph G=(V,E) all of whose equivalence classes are consistent. Let P be a conformist order produced by Lemma 3.1.3. P can be extended to a proper order $<$ for G.

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 <u>*PK graph*</u> if there is an order $<$ on the vertices which satisfies the P_4 and $2K_2$ rules.

Theorem 3.3.1. (Characterization III)

A graph G is coTT graph if and only if G is both 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 equivalence class is consistent. □

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

Algorithm 1

Step 1. Check to see if G is strongly chordal by applying Farber's Algorithm. 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 cyclic then partition the graph into smaller subgraphs and apply the algorithm recursively to form a conformist order for G as in Theorem 3.1.1.

Step 4. Extend the conformist order to a proper order as in Theorem 3.2.1.

We note that Algorithm 1 actually provides a proper order for a coTT graph G. From this order, we can obtain weights and tolerances for each vertex which satisfy the requirement for a threshold tolerance representation for G using Theorem 2.5. If we only want to check if G is coTT, we need only use Farber's Algorithm to check that G is strongly chordal, and form the P_4 and $2K_2$ equivalence classes and 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 3.1.3. The leaves of the tree are disjoint subgraphs and so we apply Algorithm 1 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 1.

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 obtained a polynomial-time recognition algorithm. We have also shown that the complements of these graphs contain both the classes of interval graphs and threshold graphs, and is contained in both the classes of strongly chordal graphs and tolerance graphs.

[BHW82] also study a generalization of threshold graphs which they call threshold signed graphs. These graphs are incomparable to coTT graphs since C_4 is in their class but not ours, and the graph in Figure 1(a) is in our class but not theirs.

Chordal graphs [Bu74, Ga78, Wa78] and strongly chordal graphs [Fa82] are also characterized in terms of intersection graphs of certain subtrees in a tree. These and other classes of graphs arising as the intersection graphs of paths in a tree are studied in [MW85]. 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 Figure 4. [CH77] 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.

ACKNOWLEDGEMENT

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

REFERENCES

[BHW85] C. Benzaken, P. L. Hammer and D. deWerra, "Threshold characterization of graphs with Dilworth number two", *Journal of Graph Theory*, to appear.

[BL76] 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**, 335-379, 1976.

[Bu74] P. Buneman, "A characterization of rigid circuit graphs," *Discrete Math.*, **9**, 205-212, 1974.

[CH77] V. Chvatal and P. L. Hammer, "Aggregation of inequalities in integer programming", *Annals of Discrete Mathematics*, **1**, 145-162, 1977.

[CN84] G. J. Chang and G. L. Nemhauser, "The k-domination and k-stability problem on sun-free chordal graphs", *SIAM Journal of Algebraic and Discrete Methods*, **5**, 332-345, 1984.

[Di61] G. A. Dirac, "On rigid circuit graphs", *Abh. Math. Sem. Univ. Hamburg*, **25**, 71-76, 1961.

[Fa82] M. Farber, "Applications of LP duality to problems involving independence and domination", Ph.D Thesis, Rutgers University, 1982.

[Fa83] M. Farber, "Characterizations of strongly chordal graphs", *Discrete Mathematics*, **43**, 173-189, 1983.

[FG65] D. R. Fulkerson and O. A. Gross, "Incidence matrices and interval graphs", *Pacific Journal of Mathematics*, **15**, 835-855, 1965.

[Ga74] F. Gavril, "The intersection graph of subtrees in trees are exactly the chordal graphs", *Journal of Combinatorial Theory B*, **16**, 47-56, 1974.

[GH64] P. C. Gilmore and A. J. Hoffman, "A characterization of comparability graphs and interval graphs," *Canad. J. Math.*, **16**, 539-548, 1964.

[Go78] M. C. Golumbic, "Threshold graphs and synchronizing parallel processes", in "*Combinatorics*" (A. Hajnal and V. T. Sos, eds.), *Colloq. Math. Soc. Janos Bolyai*, **18**, 419-428, 1978.

[Go80] M. C. Golumbic, *Algorithmic Graph Theory* and *Perfect Graphs*, Academic Press, New York, 1980.

[GM82] M. C. Golumbic and C. L. Monma, "A generalization of interval graphs with tolerances", *Proc. 13th Southeastern Conf. on Combinatorics, Graph Theory and Computing, Congressus Numeration*, **35**, (Utilatis Math., Winnepeg), 321-331, 1982.

[GMT84] M. C. Golumbic, C. L. Monma and W. T. Trotter, Jr., "Tolerance graphs", Discrete Applied Mathematics, **9**, 157-170, 1984.

[HS58] A. Hajnal and J. Surányi,"Über die auflösung von Graphen in vollständige Teilgraphen," *Ann. Univ. Sci. Budapest Eötvös Sect. Math.* 1, 113-121, 1958.

[HZ77] P. B. Henderson and Y. Zalchstein, "A graph-theoretic characterization of the PV chunk class of synchronizing primitives", *SIAM Journal of Computing*, **6**, 88-108, 1977.

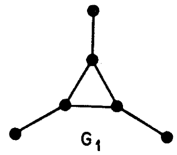
[LB62] C. G. Lekkekerker and J. C. Boland, "Representation of a finite graph by a set of intervals on the real line", *Fund. Math*, **51**, 45-64, 1962.

[MW85] C. L. Monma and V. Wei, "Intersection graphs of paths in a tree", *Journal of Combinatorial Theory B,* to appear.

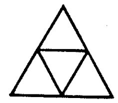
[Or77] J. Orlin, "The minimal integral separation of a threshold graph", *Annals of Discrete Mathematics*, **1**, 415-419, 1977.

[Ro70] D. J. Rose, "Triangulated graphs and the elimination process", *Journal of Math. Anal.*, **32**, 597-609, 1970.

[Wa78] J. R. Walters, "Representatives of chordal graphs as subtrees of trees," *J. Graph Theory*, **2**, 265-267, 1978.



(a) G_1 IS COTT BUT NOT INTERVAL OR THRESHOLD

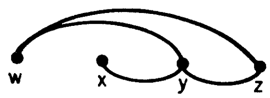


G_2

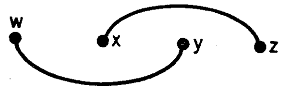
(b) G_2 IS INTERVAL TOLERANCE BUT NOT COTT

Figure 1. Example Graphs

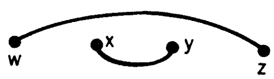
(a)



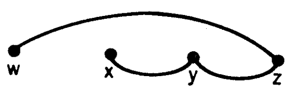
(b)



(c)



(d)



(e)

Figure 2. Forbidden Configurations in Proper Orders where $w < x < y < z$

THRESHOLD

NOT COTT

s

ers where $w < x < y < z$

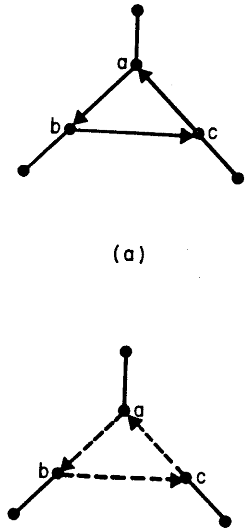


Figure 3. Cyclic Triangles

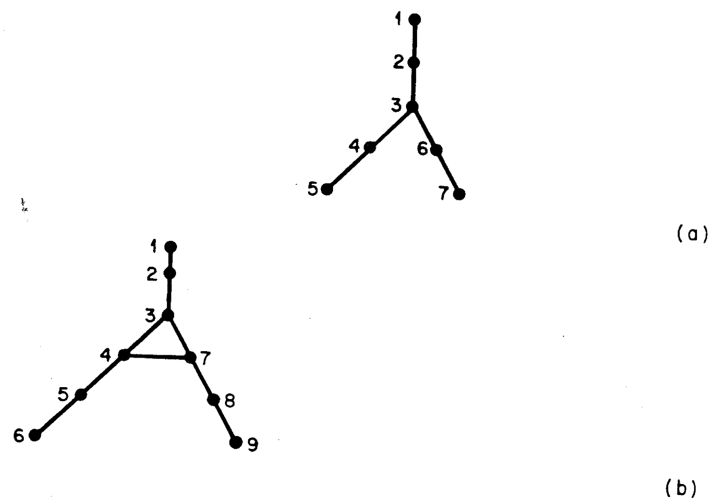


Figure 4. Forbidden Subgraphs for coTT Graphs