On Recognition of Threshold Tolerance Graphs and their Complements

Petr A. Golovach *,
Pinar Heggernes **, Nathan Lindzey ***,
Ross M. McConnell †, Vinicius Fernandes dos Santos ‡, Jeremy P. Spinrad §, Jayme Luiz Szwarcfiter ¶

Abstract. A graph $G = (V, E)$ is a threshold tolerance graph if each vertex $v \in V$ can be assigned a weight $w_v$ and a tolerance $t_v$ such that two vertices $x, y \in V$ are adjacent if $w_x + w_y \geq \min(t_x, t_y)$. Currently, the most efficient recognition algorithm for threshold tolerance graphs is the algorithm of Monma, Reed, and Trotter which has an $O(n^4)$ runtime. We give an $O(n^2)$ algorithm for recognizing threshold tolerance and their complements, the threshold tolerance (co-TT) graphs, resolving an open question of Golumbic, Weingarten, and Limouzy.

1 Introduction

Tolerance graphs are an important subclass of perfect graphs that generalizes both interval graphs and permutation graphs [8]. They have been written about extensively and they model constraints in various combinatorial optimization and decision problems [8, 9, 10]. They have a rich structure and history, and interesting relationships to other graph classes. For a detailed overview of the class, see [10].

A graph $G = (V, E)$ is threshold tolerance if each vertex $v \in V$ can be assigned a weight $w_v$ and a tolerance $t_v$ such that two vertices $x, y \in V$ are adjacent when $w_x + w_y \geq \min(t_x, t_y)$ [13]. When the tolerances of the vertices are all the same, we obtain the subclass of threshold graphs [4].

Their complements, the co-threshold tolerance graphs (co-TT graphs), have also received attention as they have an interesting interpretation as a generalization of interval graphs. They are a special case of the tolerance graphs.

A graph $G = (V, E)$ is an interval graph if and only if each vertex $v \in V$ can be assigned an interval $I_v = [a(v), b(v)]$ on the real line such that two vertices $x, y \in V$ are adjacent exactly when their corresponding intervals intersect, in which case $I = \{[a(v), b(v)] : v \in V\}$ forms an interval model of $G$. See [6, 17, 3] for surveys of the properties of this class and its relationship to other graph classes.

To illustrate the relationship of the interval graphs to the co-TT graphs, the definition can be rephrased:

* Petr.Golovach@ii.uib.no, Department of Informatics, University of Bergen, P.O.Box 7803, N-5020 Bergen, Norway.
** pinar@ii.uib.no, Department of Informatics, University of Bergen, P.O.Box 7803, N-5020 Bergen, Norway.
*** lindzey@cs.colostate.edu, Mathematics Department, Colorado State University, Fort Collins, CO, 80523-1873 U.S.A.
† rmm@cs.colostate.edu, Computer Science Department, Colorado State University, Fort Collins, CO, 80523-1873 U.S.A.
‡ vinicius@ime.uerj.br, Departamento de Matematica Aplicada, Universidade do Estado do Rio de Janeiro - UERJ, Brasil
§ spin@vuse.vanderbilt.edu, Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, U.S.A.
¶ jayme@nce.ufrj.br, Instituto de Matematica, Universidade Federal do Rio de Janeiro, Caixa Postal 2324, 20001-970 Rio de Janeiro, RJ, Brasil.
Definition 1. A graph \( G = (V, E) \) is an interval graph if and only if there exist functions \( a, b : V \mapsto \mathbb{R} \) such that:

- \( a(x) \leq b(x) \) for all \( x \in V \);
- \( xy \in E \iff a(x) \leq b(y) \wedge a(y) \leq b(x) \) for all \( x, y \in V \).

By this definition, \([a(x), b(x)]\) is the interval that represents \( x \) in the model. Relaxing the requirement that \( a(x) \leq b(x) \), gives the class of co-TT graphs:

Definition 2. \([13]\) A graph \( G = (V, E) \) is a co-TT graph if and only if there exist functions \( a, b : V \mapsto \mathbb{R} \) such that:

- \( xy \in E \iff a(x) \leq b(y) \wedge a(y) \leq b(x) \) for all \( x, y \in V \).

It is easy to see that this class is the complement of threshold tolerance graphs by setting \( a(x) = w_x \) and \( b(x) = t_x - w_x \) for all \( x \in V \) gives functions that show that the complement is a co-TT graph, and given the functions \( a() \) and \( b() \), for a co-TT graph, assigning \( w_x = a(x) \) and \( t_x = b(x) + w_x \) gives the weights and thresholds that show that its complement is a threshold-tolerance graph.

Following the notation in Golumbic, Weingarten and Limouzy \([11]\), let the blue-red partition of \( V \) given by a co-TT model be \((B, R)\), where \( B = \{x \mid x \in V \text{ and } a(x) \leq b(x)\} \) and \( R = \{x \mid x \in V \text{ and } b(x) < a(x)\} \). Given such a partition, let \( B \) be the blue vertices and \( R \) be the red vertices. The red intervals are the intervals \([b(x), a(x)]\) corresponding to red vertices and the blue intervals are the intervals \([a(x), b(x)]\) corresponding to blue vertices. Collectively, these intervals, together with their coloring, are a co-TT model.

The following is easily verified using Definition 2 (see Figure 1):

Fig. 1: Vertices that are blue in the model are black and vertices that are red in the model are gray. Though there exist models with as few as one red vertex, there are none where all vertices are blue, hence the graph is not an interval graph. The red vertices are an independent set. A red vertex is adjacent to a blue vertex if its interval is contained in the blue vertex’s interval. The pairs on the lower-right illustrates the conversion of interval endpoints to weights and thresholds that represent the complement as a threshold tolerance graph.

### Lemma 1. \([11]\) Given a co-TT model of a co-TT graph \( G \), let \((B, R)\) be its blue-red partition. Then:

- If \( \{x, y\} \subseteq B \), then \( xy \in E \iff [a(x), b(x)] \text{ and } [a(y), b(y)] \text{ intersect} \);
- If \( \{x, y\} \subseteq R \), then \( xy \notin E \);
- If \( x \in B \) and \( y \in R \), then \( xy \in E \iff [b(y), a(y)] \text{ is contained in } [a(x), b(x)] \).
It follows that the red vertices are an independent set. Figure 1 gives an example.

A chord on a cycle $C$ in a graph is an edge not on the cycle but whose endpoints are on the cycle. A graph is chordal if every cycle on four or more vertices has a chord, see, for example [6]. A chord $xy$ in an even cycle $C$ is odd when the distance in $C$ between $x$ and $y$ is odd. A graph is strongly chordal if it is chordal and every even-length of size at least six has an odd chord [5].

The following illustrates an interesting relationship between the chordal graphs, the strongly chordal graphs, the co-TT graphs and the interval graphs. A graph is chordal if and only if there is a perfect elimination ordering, which is an ordering $(v_1, v_2, \ldots, v_n)$ of its vertices such that for every vertex $v_i$, $v_i$ and its neighbors to the right form a complete subgraph. It is easily seen that ordering the vertices of an interval or co-TT graph according to left-to-right order of $b()$ in an interval or co-TT model gives a perfect elimination ordering.

A graph is strongly chordal if and only if it has a simple elimination ordering, which is an ordering $(v_1, v_2, \ldots, v_n)$ such that for each $v_i$, the neighbors of $v_i$ in $G[v_i, v_{i+1}, \ldots, v_n]$ are ordered by closed neighborhood containment. That is, if $v_j, v_k$ are two such neighbors, one of $N[v_j]$ and $N[v_k]$ is a subset of the other in $G[v_i, v_{i+1}, \ldots, v_n]$. A simple elimination is a special case of a perfect elimination ordering. Another characterization is that a graph is strongly chordal if and only if it has no strong elimination ordering, $(v_1, v_2, \ldots, v_n)$ such that whenever $v_i, v_j, v_k$ are three vertices and $i < j < k$, then $N[v_j] \subseteq N[v_k]$ in $G[v_i, v_{i+1}, \ldots, v_n]$. It is easily seen that ordering the vertices of an interval or co-TT graph according to left-to-right order of $b()$ in an interval or co-TT model also gives a strong elimination ordering. The following is an immediate consequence:

**Theorem 1.** [13] Every co-TT graph is strongly chordal.

A graph is co-TT if and only if it has a proper elimination ordering, which is an ordering $(v_1, v_2, \ldots, v_n)$ of vertices such that whenever $v_i, v_j \not\in E$, either $v_i$ is to the left of all members of $N(v_j)$ or $v_j$ is to the left of all members of $N(v_i)$. A proper elimination ordering is a simple elimination ordering and a strong elimination ordering, hence a perfect elimination ordering. It is easily seen that ordering all vertices in left-to-right order of $b()$ in an interval or co-TT model also gives this. It is shown in [13] that a graph is a co-TT graph if and only if it admits a proper elimination ordering.

To complete this taxonomy, a graph is an interval graph if and only if it admits a proper elimination ordering $(v_1, v_2, \ldots, v_n)$, whenever $v_i, v_j \not\in E$ and $i < j$, then $v_i$ is to the left of all members of $N[v_j]$. Equivalently, for comparison to proper elimination orderings, this is an ordering where $v_i$ is to the left of all members of $N[v_j]$ or $v_j$ is to the left of all members of $N[v_i]$. Ordering the vertices in left-to-right order of $b()$ in an interval model, that is, by right endpoint, gives such an ordering. Conversely, it is easy to obtain an interval model, given such an ordering, by making $j$ be the right endpoint of each vertex $v_j$ and extending the left endpoint far enough to the left to meet the right endpoint of the earliest neighbor of $v_j$. Such an ordering is a proper elimination ordering, hence a simple elimination ordering and a perfect elimination ordering.

Note that $R$ is an independent set, and that the blue intervals are an interval model of $G[B]$, hence $G[B]$ is an interval graph. A vertex $v$ of a graph is simplicial if its neighbors induce a complete subgraph. For each $r \in R$, the intervals corresponding to neighbors of $r$ contain $r$’s interval, so they have a common intersection point. Since the neighbors of $r$ are blue, $r$ is simplicial.

Henceforth, we will denote a co-TT model as $I(B, R)$, which is a set of intervals on the line, together with an implied bijection from vertices to the intervals, and where $(B, R)$ is the blue-red partition. Suppose $x$ maps to an interval $[l, r]$. If $x \in B$, $a(x)$ is implicitly $l$ and $b(x)$ is implicitly $r$, whereas if $x \in R$, $a(x)$ is implicitly $r$ and $b(x)$ is implicitly $l$.

Despite the similarities co-TT models and interval models, the best time bound for recognition of threshold tolerance and co-TT graphs until now has been $O(n^4)$ [13], whereas linear-time recognition of interval graphs has been known for some time [2].
A graph is a split graph if its vertices can be partitioned into a complete subgraph and an independent set. Golumbic, Limouzy and Weingartner [11] showed that split co-TT graphs, that is, those graphs that are both split graphs and co-TT graphs, can be recognized in \(O(n^2)\) time and a forbidden subgraph characterization for split co-TT graphs was given. We generalize this bound to recognition of arbitrary co-TT graphs. The structural insight of Section 3, developed in [11] is essential to our approach. This gives an \(O(n^2)\) bound for recognition of threshold tolerance graphs also, since it now takes \(O(n^2)\) time to recognize whether the complement of a graph is a co-TT graph.

2 Preliminaries

Two sets overlap if they intersect and neither is a subset of the other.

Given a set \(A\) of directed edges, let \(A^T\) denote the transpose \(\{(y, x) | (x, y) \in A\}\). We view undirected graph as a special case of a directed graph, where each edge \((x, y)\) consists of two arcs \((x, y)\) and \((y, x)\).

Given a binary \((0,1)\) matrix, we treat the rows and columns as bit-vector representations of sets. A row is the set of columns where the row has a 1, and similarly for columns. This allows us to apply set operations to rows or to columns, such as evaluating whether one row is a subset of another.

An undirected graph \(G = (V, E)\) is a special case of a symmetric directed graph, so we may refer to the directed edges of an undirected graph. For \(\emptyset \subset V' \subseteq V\), let \(G[V']\) denote the subgraph of \(G\) induced by \(V'\). For \(v \in V\), let the open neighborhood of \(v\), denoted \(N_G(v)\), be the set of neighbors of \(v\) in \(G\), and let its closed neighborhood, denoted \(N_G[v]\), be \(N_G(v) \cup \{v\}\). When \(G\) is understood, we may denote these \(N(v)\) and \(N[v]\).

A maximal clique of a graph is a complete subgraph that is properly contained in no other complete subgraph. Two vertices \(u\) and \(v\) are false twins if \(N(u) = N(v)\). Note that this implies that they are nonadjacent. They are true twins if \(N[u] = N[v]\), which implies that they are adjacent. The pairs of false twins in a graph are an equivalence relation, as are the pairs of true twins.

Given a collection of lists of integers from \(\{1, 2, \ldots, n\}\), whose sum of lengths is \(k\), we may sort each list by numbering the lists, sorting all the elements of all the lists in a single radix sort using list number as primary sort key and element value as secondary sort key. This takes \(O(n + k)\) time. We can sort the adjacency lists of a graph in \(O(n + m)\) time, for example. Also, we can sort the collection of the lists lexicographically in \(O(n + k)\) time even though they have different lengths [1]. The following is a consequence that we will refer to:

**Proposition 1.** It takes \(O(n + m)\) time to identify all equivalence classes of true twins, by sorting the closed neighborhoods lexicographically, and to find all equivalence classes of false twins, by sorting the open neighborhoods lexicographically.

3 Reduction to the case where \(G\) is a co-TT graph and inferring a blue-red partition

We give an \(O(n^2)\) algorithm that has the precondition that its input graph \(G\) is a co-TT graph and the postcondition that it has returned a valid co-TT model. The reason that this suffices for recognition is that such an algorithm must fail to return a valid co-TT model if and only if its input graph is not a co-TT graph, since no valid co-TT model exists if it is not. (Our algorithm sometimes returns an invalid model, and sometimes halts when it recognizes that \(G\) lacks a property that co-TT graphs have.) Given a graph \(G = (V, E)\) and co-TT model \(\mathcal{I}(B, R)\) on \(V\), it trivially takes \(O(n^2)\) time to determine whether \(\mathcal{I}(B, R)\) is a valid co-TT model of \(G\), by applying Lemma 1 to each pair of intervals and comparing the result with the corresponding adjacency-matrix entry for \(G\). We show how to implement the algorithm so that it halts in \(O(n^2)\) time, whether or not \(G\) meets its precondition.
In the rest of this paper, we assume that the precondition to the algorithm is met, that is, that $G$ is a co-TT graph, except when we analyze the running time in the case where $G$ is not a co-TT graph.

A key element of our approach is the following insight, which is given by Golumbic, Limouzy and Weingartner in [11].

**Lemma 2.** If $G$ is a co-TT graph, then there exists a co-TT model where the red vertices are the simplicial vertices that have no true twins in $G$, and the blue vertices are all others.

In the graph on the left of Figure 2, for example, the simplicial vertices are $\{d, d', e, e', f\}$. Those that have no true twins are $e, e'$ and $f$, by the lemma, these are red in some co-TT model, and $d$ and $d'$ are true twins, so these are blue in some co-TT model. The model at the bottom left illustrates this. The idea behind the proof is that if a vertex is red, then it is contained in those of its blue neighbors and it has no red neighbors. The intervals of its blue neighbors must therefore have a common intersection, and they must induce a complete subgraph. A simplicial vertex that has no true twins can be inserted as a red interval in this common intersection. If it has a true twin, then at most one of the vertex and its twin can be red, since they are adjacent. However, if one of them is blue, then the other can be modeled with a second blue interval that matches that of the first.

By this, the authors implied an obvious linear-time algorithm for finding a blue-red partition in an arbitrary co-TT graph. However, since they only addressed split co-TT graphs, they did not give it explicitly. For completeness, we give it here:

**Lemma 3.** If $G$ is a co-TT graph, it takes $O(n+m)$ time to find a blue-red partition $(B, R)$.

*Proof.* It takes $O(n+m)$ time to recognize whether a graph is chordal, the sum of cardinalities of the maximal cliques of a chordal graph is $O(n+m)$ and they take $O(n+m)$ time to find [15]. Since a co-TT graph is chordal, find its maximal cliques, in $O(n+m)$ time. A vertex is simplicial if and only if it is a member of exactly one maximal clique. It takes time proportional to the sum of cardinalities of the maximal cliques to test this on all vertices, by traversing the vertices in each maximal clique, marking them, and marking each vertex as non-simplicial if it has already been marked during traversal of another maximal clique. By Proposition 1, it takes $O(n+m)$ time to identify all equivalence classes of true twins.

This gives a reduction to finding whether a graph $G'$ has a co-TT model with a given partition $(B', R')$, where $B'$ has no true twins and the vertices in $R'$ have no false twins. The idea behind the reduction is given in Figure 2, and the reduction is given in Algorithm 1.

**Lemma 4.** The reduction of Algorithm 1 is correct, and can be implemented to run in $O(n+m)$ time whether or not the input graph is a co-TT graph.

*Proof.* If $\mathcal{I}(B, R)$ is a co-TT model of $G$, then $\mathcal{I}(B, R) \cap (B' \cup R') = \mathcal{I}(B', R')$ is a co-TT model of $G'$. Therefore, $G'$ has a co-TT model with blue-red partition $(B', R')$. Given $\mathcal{I}'(B', R')$, correctness of the construction of a co-TT model of $G$ is then immediate from Lemmas 1 and 3. The time bound given in the proof of Lemma 3 depends only on the input graph $G$ being chordal, which takes $O(n+m)$ time to determine [15]. Since all co-TT graphs are chordal, $G$ can be rejected as a co-TT graph if it is not chordal.

The motivation for the reduction is given by the following lemma, which we use to simplify the analysis:

**Lemma 5.** Let $G'$, $B'$, $R'$ be as in Algorithm 1. For any pair $\{x, y\}$ of distinct vertices, $N[x] \neq N[y]$, and for any pair $\{r, r'\} \subseteq R'$ $N(r) \neq N(r')$.

*Proof.* The red vertices of $G$ remain simplicial in $G'$. If $b$ and $b'$ are two members of $B'$, then since each true-twin equivalence class of $B'$ has only one member, $N[b] \neq N[b']$. If $r$ is a red vertex, it has no true twins in $G'$, hence it has no true twins in $G'$, and $N[r] \neq N[y]$ for any other vertex $y$. Since $r$ is the only red member of its false-twin equivalence class in $G'$, $N(r) \neq N(r')$ for any other red vertex $r'$.
Fig. 2: The reduction of Algorithm 1. A blue-red partition is given by Lemma 2. All but one member of each equivalence class of blue twins is eliminated. In the figure, $d'$ is eliminated from the equivalence class \{d, d'\}. All but one member of each equivalence class of red twins is eliminated. In the figure, $e'$ is eliminated from the equivalence class \{e, e'\}. If a model can be found for the reduced graph on the right, then it can be turned into a model of the original graph by duplicating intervals from the interval corresponding to the retained member of each twin equivalence class of size greater than 1.

**Data:** A co-TT graph $G$

**Result:** A co-TT model of $G$

1. Find the blue-red partition $(B, R)$ for some co-TT model of $G$ (Lemma 3);
2. $B' \leftarrow$ one representative from each equivalence class of true twins of $B$ (Proposition 1);
3. Remove any isolated vertices from $R$;
4. $R' \leftarrow$ one representative from each equivalence class of false twins of $R$ (Proposition 1);
5. $G' \leftarrow G[B' \cup R']$;
6. Find a co-TT model $\mathcal{I}'(B', R')$ of $G'$ (Algorithm 2);
7. for $b \in B \setminus B'$ do
6. Insert a blue interval for $b$ to $\mathcal{I}'$ equal to that of the representative of $b$'s true-twin class;
8. end
9. for $r \in R \setminus R'$ do
10. if $r$ is an isolated vertex then
11. Insert a red interval for $r$ that contains all blue intervals;
12. end
13. else
14. Insert a red interval for $r$ to $\mathcal{I}'$ equal to that of the representative of $r$'s false-twin class;
15. end
16. end
17. Return the resulting model $\mathcal{I}(B, R)$;

**Algorithm 1:** Co-TT-Model($G$)
Henceforth in the paper, we will let $G'$, $B'$ and $R'$ denote these elements of the reduction of Algorithm 2, and let $V' = B' \cup R'$ denote the vertices of $G'$.

4 Strongly Chordal Graphs and Chordal Bipartite Graphs

An edge-vertex incidence matrix for a graph has one row for each vertex, one column for each edge, and a 1 in row $i$, column $j$ if edge $j$ is incident to vertex $i$. A binary matrix is totally balanced if and only if it does not have as a submatrix the edge-vertex incidence matrix of a cycle of length at least three. (See Figure 3.) The augmented adjacency matrix of a graph on vertex set \{v_1, v_2, \ldots, v_n\} is the binary matrix that has a 1 in row $i$, column $j$ if $v_j \in N[v_i]$. That is, it is the result of adding 1's on the diagonal to the adjacency matrix. The bipartite adjacency matrix for a bipartite graph $G = (\{v_1, v_2, \ldots, v_j\}, \{w_1, w_2, \ldots, w_k\}, E)$ is the binary matrix that has a 1 in row $i$, column $j$ if $w_j$ is a neighbor of $v_i$.

Fig. 3: The edge-vertex incidence matrix of a cycle of length three. A one in a row and column indicates that the edge of the column is incident to the vertex of the row. Omitted entries in the figure are implicitly zeros. Rows $a$ and $b$ and columns $e$ and $g$ induce a Gamma, which is a $2 \times 2$ binary matrix that has a 0 only in its lower right corner. No permutation of rows and columns of the matrix eliminates the existence of a Gamma. The minimal matrices such that no permutation of rows and columns eliminates the existence of Gammas are the edge-vertex incidence matrices of cycles of length greater than or equal to three. It follows that if a binary matrix is totally balanced if and only if there exists a permutation of rows and columns that is Gamma-free.

Theorem 2. [5] A graph is strongly chordal if and only if its augmented adjacency matrix is totally balanced.

A bipartite graph is chordal bipartite if every cycle of length greater than or equal to six has a chord. (See [17] for a survey.)

Theorem 3. [7] A bipartite graph is chordal bipartite if and only if its bipartite adjacency matrix is totally balanced.

A doubly-lexical ordering of a matrix is an ordering of its rows and columns such that for two rows $i$ and $j$, if $k$ is the rightmost column where the rows differ, then the row that has a 1 in column $k$ is below the other, and for two columns $i'$ and $j'$, if $k'$ is the lowest row where the columns differ, then the column with the 1 in row $k'$ is to the right of the other. An $O(m \log n)$ algorithm for finding a doubly-lexical ordering of any binary matrix, given with a sparse representation, is given by Paige and Tarjan [14], where $n$ is the number of rows and columns and $m$ is the number of 1's. An $O(pq)$ variant is given by Spinrad for $p \times q$ matrices [16]. In this paper, we make use of the latter result.

Lemma 6. Let $\{V_1, V_2\}$ be a partition of vertices of a strongly chordal graph, $G$. Then the bipartite graph $H = (V_1, V_2, \{xy|x \in V_1 \text{ and } y \in V_2\})$ is chordal bipartite.

Proof. Any cycle $C$ in $H$ is a cycle of $G$. If $|C| \geq 6$, then it is a cycle that has an odd chord in $G$. Since the vertices on $C$ alternate between $V_1$ and $V_2$ around $C$, the odd chord has one end in $V_1$ and the other in $V_2$, hence it is a chord of $C$ in $H$. 

7
Theorem 4. [16] Given a $p \times q$ binary matrix, it takes $O(pq)$ time to determine whether it is totally balanced, and, if so, to determine for each ordered pair $(i, j)$ of rows whether row $i$ is a subset of row $j$.

The idea behind the proof is to use the algorithm of [16] to find a doubly-lexical ordering. From this, it is easy to identify identical rows, since they are consecutive. There is an easy linear-time test either to detect a Gamma or to determine that the matrix is Gamma-free. If it is Gamma-free, then for rows $i$ and $j$ with $i$ earlier than $j$, if the rows are not equal, then row $j$ cannot be a subset of row $i$, since $j$ has a 1 and $i$ has a 0 in the rightmost column where the two rows differ. To determine whether row $i$ is a subset of row $j$, let $h$ be the first column where row $i$ has a 1. If row $j$ has a 0 in column $h$, then row $i$ is not a subset of row $j$. Otherwise, row $i$ must be a subset of row $j$, which is seen as follows. Suppose to the contrary that it is not. Then there must be some column $h'$ where row $i$ has a 1 and $j$ has a 0. Then rows $i, j$ and columns $h, h'$ induce a Gamma, a contradiction.

The main consequence of these results for this paper is summarized as follows:

Lemma 7. In $G'$, it takes $O(n^2)$ time to find whether $N[b_1] \subset N[b_2]$ for each ordered pair $(b_1, b_2)$ of distinct vertices and whether $N(r_1) \subset N(r_2)$ for each ordered pair of distinct vertices in $R'$.

Proof. The bound for closed neighborhood containments follows from Theorems 2, 1, 4 and Lemma 5. Since $R'$ is an independent set of $G'$, the bound for open neighborhood containments follows from Theorems 3, 1, 4, Lemma 5, and an application of Lemma 6 to the bipartite graph $H = (R', B', E \cap (R' \times B'))$.

5 Finding a co-TT model $\mathcal{I}(B', R')$ of $G'$

In this section, we given an algorithm to find a co-TT model $\mathcal{I}(B', R')$ of $G'$, where $(G', B', R')$ are as defined in the reduction of Algorithm 2.

Definition 3. A set $\mathcal{I}$ of $n$ intervals on the line is in standard form if all endpoints are distinct and elements of $\{1, 2, \ldots, 2n\}$.

For example, the co-TT model of Figure 1 is in standard form.

Definition 4. Let $G = (V, E)$ be a graph. Let $A_V$ denote $\{(x, y) | x, y \in V \text{ and } x \neq y\}$. Let $\mathcal{I}$ be a set of intervals in standard form. For each vertex $x$, assign a member $I_x \in \mathcal{I}$ to $x$, so that the assignment is a bijection from $V$ to $\mathcal{I}$.

For $(x, y) \in A_V$, let its intersection type be an overlap if $I_x$ and $I_y$ overlap, a non-intersection if $I_x$ and $I_y$ are disjoint, a containment if $I_x$ contains $I_y$, and a subset relationship if $I_x$ is a subset of $I_y$. These labels are intersection labels of elements of $A_V$. Let us say that $\mathcal{I}$ realizes the assignment of these labels. Let $E_n(\mathcal{I})$ denote the set of elements of $A_V$ that are labeled as non-intersections, $E_o(\mathcal{I})$ those that are labeled as overlaps, $E_s(\mathcal{I})$ those that are labeled as subset relationships, and $E_c(\mathcal{I})$ be those that are labeled as containments. If $\mathcal{I}$ is understood, we may denote them $E_n$, $E_o$, $E_s$ and $E_c$, respectively (see Figure 4 for an illustration).

Lemma 8. [12] Given a set $V$ and an arbitrary assignment of intersection labels to the elements of $A_V$, it takes $O(n^2)$ time to find a set of intervals in standard form that realizes the labeling, or else to determine that no such set of intervals exists.

The bound given in [12] is actually $O(n + k)$ where $k$ is the number of elements of $A_V$ that are not in $E_n$. When the labeling is for an interval model of a graph $G$, construction of the interval model takes $O(n + m)$ time, since the edges of $G$ are those elements of $A_V$ that are not in $E_n$. Unfortunately, this bound does not apply to co-TT models, which assign a variety of labels to elements of $A_V$ that are not edges of $G$. 

8
Suppose the preconditions are met, but that a red extension fails to exist. Then

**Proof.** Suppose the preconditions are met, but that a red extension fails to exist. Then \( r_r < l_r \) for some \( r \in R' \). Since \( I \) is an interval model of \( G'[B'] \), this implies that \( r \) has two blue neighbors that are nonadjacent to each other, and \( r \) fails to be simplicial, a contradiction.

**Proposition 2.** Suppose \( \mathcal{I}(B', R') \) is a co-TT model of \( G' \). Then the red extension \( \mathcal{I}'(B', R') \) of \( \mathcal{I}(B', R') \) is a co-TT model of \( G' \).

By Proposition 2, any normal co-TT model can be turned into a co-TT model by replacing the model with the red extension of its blue intervals:

**Definition 6.** If \( G' \) is a co-TT graph, then a co-TT model \( \mathcal{I}(B', R') \) of \( G' \) is normalized if \( \mathcal{I}(B', R') \) is the red extension of \( \mathcal{I}(B', R')[B'] \).

The algorithm for assigning the intersection labels to \( A_V \) is given as Algorithm 2, and gives a summary of the roles of the lemmas that follow in this section.

**Lemma 10.** There exists a co-TT model \( \mathcal{I}(B', R') \) of \( G' \) such that for each \( (b_1, b_2) \in A_{B'} \):

- \( (b_1, b_2) \in E_n \) if \( b_1 \) and \( b_2 \) are nonadjacent;
- \( (b_1, b_2) \in A_s \) and \( (b_2, b_1) \in A_c \) if \( N[b_1] \subset N[b_2] \);
- \( (b_1, b_2) \in E_o \) if \( b_1 \) and \( b_2 \) are adjacent but neither \( N[b_1] \subset N[b_2] \) nor \( N[b_2] \subset N[b_1] \).

**Proof.** Let \( \mathcal{I}'(B', R') \) be a co-TT model, and let \( I_1 \) and \( I_2 \) be the intervals for \( b_1 \) and \( b_2 \), respectively, in \( \mathcal{I}'(B', R') \).

If \( b_1 \) and \( b_2 \) are nonadjacent, then since they are elements of \( B' \), \( I_1 \) and \( I_2 \) do not intersect in \( \mathcal{I}'(B', R') \), by Lemma 1. This establishes the first claim.

For the second claim, suppose \( N[b_1] \subset N[b_2] \). If \( N[b_1] \subset N[b_2] \), then if the right endpoint of \( I_1 \) lies to the right of the right endpoint of \( I_2 \) in \( \mathcal{I}(B', R') \), in violation of the claim, the
Fig. 5: The red extension of a blue model. Given a blue model (solid lines) and the adjacencies of red vertices to blue vertices, we stretch each red interval as far as we can without losing any of its blue neighbors. The vertex \( f \) has \( \{a, b, c\} \) as neighbors, so its red interval is made to fit just inside the common intersection of intervals \( a, b, \) and \( c \). The same approach is used for other red vertices. When left endpoints of red intervals coincide, we order them in descending order of right endpoint, as in the example of \( \{f, g\} \) and when right endpoints coincide, we order them in descending order of left endpoint, as in the case of \( \{f, h, i\} \). Sorting all endpoints according to this ordering gives a new model in standard form. The result might not be a co-TT model: if the neighbors of \( f \) were \( \{a, b\} \), it would have the same red interval in the red extension, falsely representing \( f \) as a neighbor of \( c \). This happens when some other blue model of \( G'[B'] \) is has a co-TT model as its red extension.
Data: The graph $G' = (V', E')$ is a co-TT graph with blue-red partition $(B', R')$, $B'$ has no true twins, and $R'$ has no isolated vertices or false twins.

Result: A labeling of elements of $A_{V'}$ with their intersection types in a co-TT model of $G'$.

1. for $(x, y) \in A_{V'}$ do
   2. Find whether $N[x] \subset N[y]$ (Lemma 7);
   3. end
4. for $(r_1, r_2) \in A_{R'}$ do
   5. Find whether $N(x) \subset N(y)$ (Lemma 7);
   6. end
7. for $(b_1, b_2) \in A_{B'}$ (Lemma 10) do
   8. if $b_1b_2 \notin E'$ then
   9. Assign $(b_1, b_2)$ to $E_n$;
10. end
11. else if $N[b_1] \subset N[b_2]$ then
12. Assign $(b_1, b_2)$ to $A_s$ and $(b_2, b_1)$ to $A_c$;
13. end
14. end
15. for $(b_1, b_2) \in A_{B'}$ do
16. if $(b_1, b_2)$ has not already been assigned then
17. Assign $(b_1, b_2)$ to $E_o$;
18. end
19. end
20. Construct an interval model $I_B$ of $G[B']$ realizing these labels (Lemma 8);
21. Let $I'$ be the red extension of $I_B$;
22. for $(r_1, r_2) \in A_{R'}$ do
23. if $N[r_1] \subset N[r_2]$ then
24. Assign $(r_1, r_2)$ to $A_c$ and $(r_2, r_1)$ to $A_s$ (Lemma 11);
25. end
26. else if $(r_1, r_2) \in E_n(I')$ then
27. Assign $(r_1, r_2)$ to $E_n$ (Lemma 12);
28. end
29. end
30. for $(b, r) \in B' \times R'$ do
31. if $br \in E'$ then
32. Assign $(b, r)$ to $A_c$ and $(r, b)$ to $A_s$ (Lemma 1);
33. end
34. else if $(b, r) \in A_s(I')$ then
35. Assign $(b, r)$ to $A_s$ and $(r, b)$ to $A_c$ (Lemma 12);
36. end
37. else if $(b, r) \in E_n(I')$ then
38. Assign $(b, r), (r, b) \in E_n$ (Lemma 12);
39. end
40. end
41. for $(x, y) \in A_{V'}$ do
42. if $(x, y)$ has not been assigned then
43. Assign $(x, y)$ to $E_o$;
44. end
45. end
46. Apply Lemma 8 to find a co-TT model $I(B', R')$ of $G'$;

Algorithm 2: Co-TT-Model($G', B', R'$)
right endpoint of $I_1$ can be moved to be just to the left of the right endpoint of $I_2$ without causing $b_1$ to lose any neighbors of $b_2$ in the represented graph. Since $N[b_1] \subset N[b_2]$, this does not cause $b_1$ to lose any neighbors in the represented graph. By symmetry, a left endpoint of $b_2$ that lies outside of $b_1$’s interval can be brought inside it without affecting the represented graph. Now $(b_1, b_2)$ conforms to the second claim. Let us call this operation a \textit{contraction}.

Number the blue vertices $(b_1, b_2, \ldots, b_k)$ in descending order of neighborhood size. For $i$ from 1 to $k$, perform a contraction on each vertex $b_i$ such that $N[b_i] \subset N[b_j]$ but whose interval is not contained in that of $b_i$. This brings $(b_i, b_j)$ into conformity with the second claim. By induction on $i$, the pairs that are in violation of the second claim are confined to $(b_{i+1}, b_{i+2}, \ldots, b_k)$ after iteration $i$. There can be no violation of the second claim after iteration $k$.

For the third claim, since $b_1$ and $b_2$ are adjacent, $N[b_1] \cap N[b_2]$ is nonempty. Since there are no true blue twins in $B'$, $N[b_1] \neq N[b_2]$. Therefore, $N[b_1] \not\subseteq N[b_2]$ and $N[b_2] \not\subseteq N[b_1]$ implies that $N[b_1]$ and $N[b_2]$ overlap. Since $b_1$ and $b_2$ are adjacent and blue, $I_1$ intersects $I_2$ by Lemma 1. If $x \in N[b_1] \cap N[b_2]$, then if $x$ is red, $I_1$ contains $x$’s interval but $I_2$ does not. $I_1$ is not contained in $I_2$. Similarly, $I_2$ is not contained in $I_1$, $I_1$ and $I_2$ intersect, but neither is contained in the other, so $(b_1, b_2) \in E_n(I'(B', R'))$.

By Lemmas 8 and 10, we may now find an assignment $I_B$ of intervals to $B'$ such that the intersection types are the same as those of $I(B', R')[B']$ for some co-TT model $I(B', R')$ of $G'$ in $O(n^2)$ time. Note that $I_B$ and $I'(B', R')[B']$ are both interval models of $G'[B']$. Since there may be many interval models satisfying these intersection types, it is not necessarily the case that $I_B = I'(B', R')[B']$ for any co-TT model $I'(B', R')$ with blue-red partition.

Henceforth, as in Algorithm 2, we will let $I_B$ denote this interval model of $G'[B']$.

Lemma 11. If $I(B', R')$ is a normalized co-TT model of $G'$, then for distinct members $r_1, r_2 \in R'$, $(r_1, r_2) \in A_s(I(B', R'))$ if and only if $N(r_2) \subset N(r_1)$.

Proof. Let $I_1$ be the interval for $r_1$ and $I_2$ be the interval for $r_2$ in $I(B', R')$.

Since $R'$ has no isolated vertices, neither $N(r_1)$ nor $N(r_2)$ is empty. If $(r_1, r_2) \in A_s(I(B', R'))$, that is, that $I_1 \subset I_2$, then any blue interval that contains $I_2$ contains $I_1$. By Lemma 1, $N(r_2) \subset N(r_1)$. Since $R'$ has no false twins, $N(r_2) \subset N(r_1)$.

Suppose that $N(r_1) \subset N(r_2)$, but $r_2$’s interval does not contain $r_1$’s in $I(B', R')$, contradicting the claim. Then either $I_1$’s right endpoint lies to the right of $I_2$’s, or $I_1$’s left endpoint lies to the left of $I_2$’s. Assume without loss of generality that $I_1$’s right endpoint lies to the right of $I_2$’s. Since $I'(B', R')$ is the red extension of $I(B', R')[B']$, $I_1$’s right endpoint lies in the consecutive block of red endpoints to the left of the leftmost right endpoint of neighbors of $r_1$. Since $N(r_1) \subset N(r_2)$, the right endpoint of $I_2$ cannot lie to the right of this endpoint by Lemma 1. Since it lies to the right of the right endpoint of $r_1$, it must lie in the same block of red right endpoints as the right endpoint of $r_1$. By the way right endpoints in such a block are ordered in a right extension, this implies that $|N(r_2)| < |N(r_1)|$, a contradiction.

Though $I_B$ realizes the intersection types of $I(B', R')[B']$ for some co-TT model $I(B', R')$, it may not be the case that $I_B = I'(B', R')[B']$ for any co-TT model $I'(B', R')$ of $G'$. There may be many interval models of $G'[B']$ that realize these intersection types. Therefore, the red extension of $I_B$ might not be a co-TT model. We can nevertheless use it to derive some of the intersection types in $I'(B', R')$:

Lemma 12. Let $I(B', R')$ be a normalized co-TT model of $G'$ whose intersection types among the blue vertices are the same as those given by $I_B$. Let $I'$ be the red extension of $I_B$.

1. For $r_1, r_2 \in R'$, $(r_1, r_2) \in E_n(I')$ if and only if $(r_1, r_2) \in E_n(I(B', R'))$:
2. For \( r \in R', b \in B', (b, r) \in E_n(I') \) if and only if \((b, r) \in E_n(I(B', R')), \) and \((b, r) \in A_s(I') \) if and only if \((b, r) \in A_s(I(B', R')).\)

Proof. For the first claim, suppose first that \( N(r_1) \cup N(r_2) \) is not a complete subgraph of \( G'. \) Since \( r_1 \) and \( r_2 \) are each simplicial, \( r_1 \) has a neighbor \( x \) and \( r_2 \) has a neighbor \( y \) such that \( x \) and \( y \) are nonadjacent. Since \( x, y \in B' \) and \( I'[B'] = I_B \) and \( I(B', R')[B'] \) are interval models of \( G'[B'] \), \( x \) and \( y \) have disjoint intervals in both models. By the construction of the red extension, \( r_1 \)'s interval is contained in \( x \)'s interval in both models and \( r_2 \)'s interval is contained in \( y \)'s interval in both models. Therefore, \( r_1 \)'s and \( r_2 \)'s intervals are disjoint in both models, so \((r_1, r_2) \in E_n(I') \) and \((r_1, r_2) \in E_n(I(B', R')).\) Now, suppose to the contrary that \( N(r_1) \cup N(r_2) \) is a complete subgraph. Since \( I'[B'] \) and \( I(B', R')[B'] \) are both interval models of \( G'[B'] \), and the intervals corresponding to \( N(r_1) \cup N(r_2) \) have a common intersection in both models. By the definition of the red extension, \( r_1 \) and \( r_2 \) intersect in this common intersection in \( I' \), which is the red extension of \( I_B, \) and in \( I(B', R'), \) which is the red extension of \( I((B', R')[B'] \). Therefore, \((r_1, r_2) \notin E_n(I') \) and \((r_1, r_2) \notin E_n(I(B', R')).\)

For the first part of the second claim, suppose first that \( N(r) \subseteq N(b) \). It follows that \( b \)'s interval cannot intersect \( r \)'s interval in \( I' \) or in \( I(B', R'), \) since all neighbors of \( r \) contain these intervals in both models. Therefore, \((r, b) \in E_n(I') \) and \((r, b) \in E_n(I(B', R')).\) Now, suppose to the contrary that \( N(r) \subset N(b) \). In both models, \( b \)'s right endpoint must end to the right of the rightmost left endpoint of neighbors of \( r \). Since both models are red extensions, \( r \)'s left endpoint lies to the left of the left endpoint of any blue interval with this property, hence to the left of \( b \)'s right endpoint. By a symmetric argument, \( r \)'s right endpoint lies to the right of \( b \)'s left endpoint. Therefore, \( r \)'s interval intersects \( b \)'s in both models, so \((b, r) \notin E_n(I') \) and \((b, r) \notin E_n(I(B', R')).\)

For the second part of the second claim, suppose \((b, r) \in A_s(I') \). Then for each neighbor \( b' \) of \( r, (r, b') \in A_s(I') \) and \((r, b') \in A_s(I(B', R')).\) Since they are both red extensions. By transitivity of \( A_s(I') \), \((b, b') \in A_s(I') \). Since both models have the same intersection labels among pairs of blue vertices, \((b, b') \in A_s(I(B', R')).\) Therefore, the left endpoint of \( b \)'s interval in \( I(B', R') \) lies to the right of the rightmost left endpoint of a neighbor of \( r \). By the construction of the red extension, \( r \)'s left endpoint lies to the left of the left endpoint of any such blue interval, hence to the left of the left endpoint of \( b \)'s interval. Similarly, its right endpoint lies to the right of the right endpoint of \( b \)'s interval, and \((b, r) \in A_s(I(B', R')).\) The proof of the converse is identical, except for reversal of the roles of \( I' \) and \( I(B', R'), \) since it makes use only of the fact that they are both red extensions and not that \( I(B', R') \) is a co-TT model.

**Lemma 13.** Algorithm 2 is correct.

Proof. Since every red vertex in a co-TT model is simplicial, the preconditions imply that every vertex in \( R' \) is simplicial.

By Lemma 10, the labeling of intersection types conducted by the \texttt{for} loop on blue pairs is consistent with those in a co-TT model \( I(B', R'). \) Therefore, \( I_B \) gives the same intersection labels as \( I_s(I(B', R')[B'] \) does. This is true also for the red extension \( I_2(B', R') = I(B', R') \cup I_3(B', R') \cup I_5(B', R') \cup I_7(B', R') \), which, by Proposition 2 and Definition 6 is a normalized co-TT model of \( G'. \) By Lemmas 11 and 12, those members \((x, y) \) of \( A_{V'} \) such that at least one of \( x \) and \( y \) is a member of \( R' \) are assigned to \( E_{n}, A_{s} \) or \( A_{V'} \) in the next two loops if and only if they have those intersection types in \( I_2(B', R') \). Since \( \{E_{n}(I_2(B', R')) \cup A_{s}(I_2(B', R')) \cup A_{V'}(I_3(B', R')) \cup E_{n}(I_7(B', R')) \} \) is a partition of \( A_{V'}, \) any elements not yet assigned must belong to \( E_{n}(I_2(B', R')) \), and the final loop correctly assigns these.

The intersection labels assigned to \( A_{V'} \) are those of \( I_2(B', R') \). The set \( I_3 \) of intervals given by Lemma 8 has these intersection types. Since \( I_2(B', R') \) is a co-TT model, so is \( I_3(B', R') \), and this is the model returned by the algorithm.

**Lemma 14.** Algorithm 2 can be implemented to take \( O(n^2) \) time even when \((G', B', R') \) does not meet the preconditions.
Proof. The application of Lemma 7 requires that $G'$ be strongly chordal. This can be checked before the lemma is applied, by Theorems 2 and 4, and $G'$ can be rejected as a co-TT graph if it fails this test, by Theorem 1.

Otherwise, the lemma gives the required neighborhood containments, whether or not $G$ is a co-TT graph, in $O(n^2)$ time. The loop at Line 7 takes $O(n^2)$ time, whether or not $G$ is a co-TT graph. Lemma 8 either gives a set of $I_B$ of intervals that realizes this labeling, or determines that none exists, in $O(n^2)$ time. By Lemma 10, such a set exists if $(G', B', R')$ meets the precondition, so $(G', B', R')$ can be rejected as failing to meet the preconditions. By Lemma 10, the intersection types of $I_B$ are the same as they are for some co-TT model $I(B', R')$ if $(G', B', R')$ meets the preconditions.

Constructing a red extension $I'$ of $I_B$ or determining that none exists takes $O(n^2)$ time by elementary methods. By Lemma 9, there is a red extension of $I_B$ if the preconditions are met, so if there is no red extension, $(G', B', R')$ can be rejected as not meeting the precondition.

Otherwise the time required for the remaining loops do not depend on any additional assumptions about the inputs, and they takes $O(n^2)$ time. The final application of Lemma 8 takes $O(n^2)$ time whether or not it succeeds in producing a set of intervals that realizes the labeling.

**Theorem 5.** Recognition of threshold tolerance and co-TT graphs takes $O(n^2)$ time.

Proof. The problems reduce to each other in $O(n^2)$ time, so we show the result for co-TT graphs. Let $G$ be a graph passed to Algorithm 1. Whether or not $G$ is a co-TT graph, the algorithm halts in $O(n^2)$ time, by Lemmas 4 and 14. If $G$ is a co-TT graph, it returns a co-TT model of $G$ by Lemmas 4 and 13. If $G$ is not a co-TT graph, it produces an incorrect co-TT model, since no co-TT model of $G$ exists, or else it halts without producing one. If it halts without producing one, $G$ can be rejected as a co-TT graph. If the algorithm produces a co-TT model, it takes $O(n^2)$ time to check whether it is a valid co-TT model for $G$, and if it is, $G$ can be accepted, and if it is not, it can be rejected, since this only happens when $G$ is not a co-TT graph.

References


