Recognizing Bipartite Unbounded Tolerance Graphs in Linear Time

Arthur H. Busch*
art.busch@notes.udayton.edu
Department of Mathematics
University of Dayton
Dayton, OH 45469-2316

Garth Isaak
gi02@lehigh.edu
Department of Mathematics
Lehigh University
Bethelem, PA 18015

March 1, 2007

Abstract

A graph $G = (V, E)$ is a tolerance graph if each vertex $v \in V$ can be associated with an interval of the real line I_v and a positive real number t_v in such a way that $(uv) \in E$ if and only if $|I_v \cap I_u| \geq \min(t_v, t_u)$. No algorithm for recognizing tolerance graphs in general is known. In this paper we present an $O(n+m)$ algorithm for recognizing tolerance graphs that are also bipartite, where n and m are the number vertices and edges of the graph, respectively. We also give a new structural characterization of these graphs based on the algorithm.

1 Introduction and Notation

A graph $G = (V, E)$ consists of a set V, called vertices and a collection E of edges, which are unordered pairs of elements of V. We assume throughout this paper that our graphs are simple and finite. In other words, $|V|$ is always finite, and E is a set which contains no edge of the form (vv). The *order* of G is $|V|$ and we will denote this throughout the paper as n. Similarly, the *size* of G is $|E|$ which we will denote by m. A graph is a *tree* if it contains no cycles, and a tree in which there is at most one vertex incident with multiple edges will be called a *star*. A graph is bipartite when the vertex set can be partitioned into two sets so that no edge connects two vertices from the same set. When G is a bipartite graph, we will represent a bipartition of V as V_x and V_y, with $n_x = |V_x|$ and $n_y = |V_y|$. When $G = (V_x, V_y, E)$ is bipartite with $V_x = \{x_1, \dots x_{n_x}\}$ and $V_y = \{y_1, \dots, y_{n_y}\}$, we will use $A(G)$ to denote the *reduced adjacency matrix* of G. This is the $n_x \times n_y$ matrix with $a_{ij} = 1$ if $(x_i y_j) \in E$ and $a_{ij} = 0$ otherwise.

*Research for this article was performed while the author was a Visiting Assistant Professor at Lehigh Univeristy

1.1 Tolerance graphs

Tolerance graphs were introduced in 1982 by Golumbic and Monma [7] to model certain scheduling problems. A graph $G = (V, E)$ is a *tolerance graph* if each vertex $v \in V$ can be associated with an interval of the real line I_v and a positive real number t_v in such a way that $(uv) \in E$ if and only if $|I_v \cap I_u| \geq \min(t_v, t_u)$. The collection $\langle \mathcal{I}, t \rangle$ of intervals and tolerances is called a *tolerance representation* of the graph G. A tolerance representation is called *bounded* when $|I_v| \leq t_v$ for every $v \in V$, and when G has such a bounded tolerance representation, we will say that G is a *bounded tolerance graph.*

Note that some authors (see [3], [6] and [15]) have studied a class of graphs that they call "bipartite tolerance graphs" but which is properly contained in the intersection of the classes of tolerance graphs and bipartite graphs (the graph T_2 in Figure 1 is a separating example, as it is both bipartite and a tolerance graph, but is not a "bipartite tolerance graph" as defined in [6]). This smaller class of graphs was shown to be equivalent to bipartite permutation graphs in [3] and [15], and it follows from a theorem of Langley [11] that the class of bipartite permutation graphs is equivalent to bipartite bounded tolerance graphs. As a result, we will follow the convention used in [9], and we will use the phrase *bipartite tolerance graph* for the intersection of tolerance graphs and bipartite graphs, and the phrase *bipartite bounded tolerance graph* for the smaller class that is equivalent to bipartite permutation graphs.

Additional background and results on tolerance graphs can be found in the recent book by Golumbic and Trenk [9]. Although tolerance graphs and related topics have been studied extensively, the problem of characterizing tolerance graphs remains open, as does tolerance graph recognition [9]. It was shown in [10] that every tolerance graph has a polynomial sized integer representation, and hence Tolerance Graphs recogniion is in NP. However, this result gives no information on how to construct an algorithm that recognizes when a graph has a tolerance representation.

The class of cycle free tolerance graphs was characterized in [8].

Theorem 1.1 (Golumbic, Monma and Trotter, [8])**.** *Let T be a tree. Then T is a tolerance graph if and only if T contains no induced subgraph isomorphic to T_3, in Figure 1.*

For bipartite graphs which contain cycles, Busch [5] gave the following characterization.

Theorem 1.2 (Busch [5])**.** *A bipartite graph G is a tolerance graph if and only if there exists a set of consecutively ordered stars $S_1, S_2, \ldots, S_t$ which partition the edges of G.*

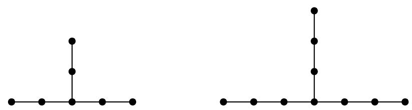

Figure 1: The trees T_2 and T_3

1.2 Asteroidal triples and consecutive orderings

We will call a collection of sets $\mathcal{U}$ *consecutively orderable* if the sets can be indexed $U_1, U_2, \dots U_k$ so that whenever $x \in U_i \cap U_k$ then $x \in U_j$ for every $i \leq j \leq k$. A collection of sets together with such an ordering will be referred to as *consecutively ordered.* In this paper, the collections of sets will generally be subsets of the vertex set of a graph and in order to conserve notation we will say that a set of subgraphs $G_1, \dots G_k$ is consecutively ordered when $V(G_1), \dots, V(G_k)$ is consecutively ordered.

A $(0,1)$-matrix M has the *consecutive* 1*'s property for rows* if the columns of M can be permuted in such a way that the 1's in every row occur consecutively. Analogously, a matrix M has the *consecutive* 1*'s property for columns* if the rows of M can be permuted in such a way that the 1's in every column occur consecutively. When M is the reduced adjacency matrix of a bipartite graph, a consecutive ordering of the columns or rows represents an ordering of either V_x or V_y such that the collection of neighborhoods $\mathcal{N}_x = N(x_1), N(x_2), \dots, N(x_{n_x})$ or $\mathcal{N}_y = N(y_1), N(y_2), \dots, N(y_{n_y})$ is consecutively ordered.

Tucker [16] investigated when the reduced adjacency matrix $A(G)$ of a bipartite graph $G = (V_x, V_y, E)$ has the consecutive 1's property for rows or columns. In this case, a row or column of $A(G)$ represents the neighborhood of a vertex in V_x or V_y, respectively. Thus, a permutation of the rows of $A(G)$ such that the 1's in every column occur consecutively is equivalent to an ordering the vertices of $V_x = \{x_1, \dots, x_{n_x}\}$ so that the collection of sets $\mathcal{N}_x = N(x_1), N(x_2), \dots, N(x_{n_x})$ are consecutively ordered. Tucker calls bipartite graphs with this property X-*consecutive*, and analagously, a bipartite graph is Y-*consecutive* when $\mathcal{N}_y = \{N(y) \mid y \in V_y\}$ is consecutively orderable. Bipartite graphs which are either X-consecutive or Y-consecutive are known as *convex*, while bipartite graphs that are both X-consecutive and Y-consecutive are *biconvex.*

An *asteroidal triple* in a graph $G = (V, E)$ is a triple of distinct vertices v_0, v_1, v_2 with the property that for each $i = 0, 1, 2$, there is a path from v_{i+1} to v_{i+2} in G that contains no vertex adjacent to v_i (subscript addition is performed modulo 3). Tucker showed the following connection between consecutive orderings and asteroidal triples.

Theorem 1.3 (Tucker [16])**.** *A bipartite graph $G = (V_x, V_y, E)$ is X- consecutive if and only if G has no asteroidal triple contained in V_x. Similarly G is Y-consecutive if and only if G has no asteroidal triple contained in V_y.*

Algorithms which determine if a matrix has the consecutive 1's property for rows form the basis for the first linear recognition algorithm for interval graphs, due to Booth and Lueker [1], and closely related algorithms also recognize convex graphs in linear time (although such algorithms generally avoid using adjacency matrices to preserve linear running times even for sparse graphs). Algorithms to identify the consecutive 1's property of a matrix can also easily be used to determine when a collection of subgraphs $\mathcal{G} = G_1 \dots G_t$ of a given graph G is consecutively orderable. We simply construct the "vertex-graph incidence matrix" $M_{n\times t} = [m_{ij}]$ which has $m_{ij} = 1$ if the vertex v_i is contained in $V(G_j)$ and $m_{ij} = 0$ otherwise. Then $\mathcal{G}$ is consecutively orderable if and only if M has the consecutive 1's property for rows. Thus, when $\mathcal{G}$ is part of the input, and $\mathcal{G}$ is a collection of stars which partition the edges of G, this can be used to show that G is a tolerance graph using Theorem 1.2. However, a tolerance graph G generally has many star partitions (the set E, for example), not all of which can be consecutively ordered. As a result, the above procedure cannot be used to decide if an arbitrary bipartite graph is a tolerance graph. In the following section, we characterize bipartite graphs whose edges can be partitioned into sets which induce stars which are consecutively orderable. We call such a partition a *consecutive star partition* (CSP), and in the process obtain a conceptually simple linear time algorithm ($O(n+m)$) that recognizes the class of bipartite tolerance graphs.

2 Bipartite Tolerance Graphs

We begin with some basic observations about consecutive star partitions and tolerance graphs. Throughout this section we will denote a consecutive star partition (CSP) of a graph G as $\mathcal{S} = S_1, S_2, \dots, S_t$, where each S_i is a star, and we will call t the *length* of $\mathcal{S}$. We will denote the vertex and edge set of the star S_j as $V(S_j)$ and $E(S_j)$, respectively. If S_i is a single edge, we will arbitrarily designate one endpoint of this edge as c_i. Otherwise, let c_i be the unique central vertex of S_i.

Observation 2.1. *Let $G = (V_x, V_y, E)$ be a connected bipartite tolerance graph with CSP $\mathcal{S} = S_1, S_2, \dots, S_t$. Then $V(S_i) \cap V(S_{i+1})$ is a cut-set of G for each $1 \leq i < t$.*

Observation 2.2. *Let G be a 2-connected bipartite graph. Then G is a tolerance graph if an only if G is convex.*

Observation 2.2, together with the hereditary property of bipartite tolerance graphs, shows that every 2-connected subgraph of a bipartite tolerance graph must be convex. Recall that a *block* of a graph G is a maximal subgraph of G with no cut-vertex. It is easy to see that every block of a connected graph is either 2-connected, a cut-edge or an isolated vertex (in the trivial case where $G = K_1$). We will define the *boundary* of a block B, denoted $\mathcal{B}(B)$ as the set of vertices in B with $N(v) \nsubseteq B$. In other words, the boundary of a block B is the set of cut-vertices of G that are also in B. If P_G is the set of pendant vertices of G, we will partition the boundary of B into two sets $\mathcal{B}^1(B) = \{v \in \mathcal{B}(B) \mid N(v) \setminus B \subseteq P_G\}$ and $\mathcal{B}^2(B) = \mathcal{B}(B) \setminus \mathcal{B}^1(B)$.

Let B be a block of a graph G. We will define B' as the graph induced by the vertices of B, together with the vertices in P_G adjacent to B. We then define a graph H_B from B' by adding two new vertices v' and v'' to B' for each vertex $v \in \mathcal{B}^2(B)$, along with the edges (vv') and $(v'v'')$. Note that H_B is an induced subgraph of G, and that if $\mathcal{B}^2(B) = \emptyset$ then $B = G$.

Our algorithm is based on the following Lemma, which is a slight extension of Observation 2.2.

Lemma 2.3. *If G is a bipartite tolerance graph, then for every block B of G, H_B is convex.*

The next two Lemmas describe the structure how the blocks of a bipartite tolerance graph are arranged, which leads to a procedure for combining the CSPs for each B' into a CSP for the graph G.

Lemma 2.4. *If G is a bipartite tolerance graph and B is a 2-connected block of G with $|\mathcal{B}^2(B)| \geq 2$, then $\mathcal{B}^2(B) = \{u, v\}$ and B' has a CSP with that begins with a star containing u and ends at a star containing v.*

As a corollary, we note a direct consequence of the contrapositive of the above result.

Corollary 2.5. *If G is a bipartite graph and B is a 2-connected block of G with $|\mathcal{B}^2(B)| > 2$, then H_B is not convex, and G is not a tolerance graph.*

Lemma 2.6. *If G is a bipartite tolerance graph, and B is a block of G with $\mathcal{B}^2(B) = \{v\}$ and v is at distance two or less from every other vertex of B', then B' has a CSP such that v is contained in every star.*

3 Class Hierarchies

In this section, we consider how various sub-classes of chordal bipartite graphs relate to the class of bipartite tolerance graphs and the implications of these relationships on the problem of recognizing bipartite tolerance graphs. We begin by noting some basic inclusions from [3]. Recall that the class of bipartite permutation graphs is equivalent to the class which we denote as bipartite bounded tolerance graphs.

(permutation ∩ bipartite) ⊂ biconvex ⊂ convex ⊂ chordal bipartite

As noted in [2], both biconvex and convex graphs can be recognized in linear time by using PQ-trees or other algorithms for the consecutive ones property, and bipartite bounded tolerance graphs can also be recognized in linear time [15]. The class of chordal bipartite graphs can be recognized in polynomial time, but no linear time algorithm is presently known [14].

Next, we note a refinement of the above hierarchy due to a combination of the results of Brown [4], Busch [5], Müller [12], and Sheng [13].

convex ⊂ (probe interval ∩ bip.) ⊂ (tolerance ∩ bip.) ⊂ interval bigraph

Although convex graphs may be recognized in linear time, the best known algorithms for the class of bipartite probe-interval graphs and for the class of interval bigraphs are polynomial. In the case of 2-connected bipartite graphs, Observation 2.2 indicates that we have **convex** = **tolerance ∩ bipartite**, and so in this restricted case, the first two inclusions above become equality. This equivalence leads us to consider blocks. By considering the blocks of a graph, and how they are arranged, we can utilize the linear time recognition of convex graphs to give a conceptually simple linear time recognition algorithm for all bipartite tolerance graphs in the next section.

Furthermore, it is easy to show that within the subclass of 2-edge-connected bipartite graphs, **convex** = **probe-interval ∩ bipartite**. This equality suggests a similar approach to the one we take below may provide a linear time algorithm for the class of bipartite probe-interval graphs. It is less certain that our approach can be extended to give a linear time recognition algorithm for the class of interval bigraphs or the class of chordal bipartite graphs.

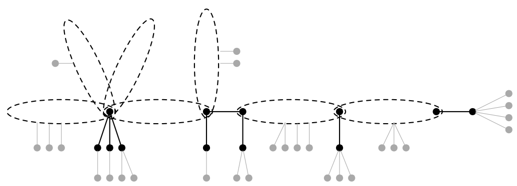

Figure 2: The block structure of $G - P_G$ for a bipartite tolerance graph G. The dashed ovals represent 2-connected blocks and the gray vertices are pendant in G.

4 An algorithm to recognize bipartite tolerance graphs

In broad terms, for a bipartite tolerance graph G, the lemmas in Section 2 impose a structure on the blocks of $G - P_G$, as well as a structure on the arrangement of those blocks. We illustrate this informally in Figure 2. After removing the pendant vertices, Corollary 2.5 implies that the blocks of $G - P_G$ can be arranged in a nearly linear structure. To make this notion more precise, we now present an algorithm that recognizes bipartite tolerance graphs.

Algorithm 1 Determine if G is a bipartite tolerance graph.

Require: G is a connected, bipartite graph
Find P_G, the set of all pendant vertices of G.
Find all the blocks of $G - P_G$ and construct the block-cutpoint graph T.
for all blocks B of $G - P_G$ **do**
    Construct B' and H_B
    if H_B is not convex **then**
        return false
    else
        if $d_T(B) = 1$ and $\epsilon_{B'}(c) \leq 2$ for the unique cut-vertex $c \in B$ **then**
            Delete B from T
        end if
    end if
end for
if T is a path **then**
    return true
else
    return false
end if

Theorem 4.1. *Algorithm 1 is correct, and runs in $O(n + m)$ steps.*

Proof. First, we show that the algorithm is correct. If $G - P_G$ contains a block B such H_B is not convex, then G is not a tolerance graph by Lemma 2.3. If H_B is convex for every block B, but T is not a path after all applicable blocks have been deleted, then T contains a vertex of degree three or more. This vertex does not represent a block of $G - P_G$, since for such a block B, $|\mathcal{B}^2(B)| \geq 3$ and so H_B is not convex by Corollary 2.5. So this vertex in T represents a cut-vertex v, and v is at the end of at least three paths of length three. Furthermore, the edges of these paths incident with v are each in different blocks of G. Thus, G contains an induced subgraph isomorphic to T_3, and hence is not a tolerance graph by Theorem 1.1. In all other cases, the algorithm returns true. In such cases, either G is a star and hence obviously a tolerance graph, or we can construct a CSP for G as follows:

Since each block of T has a corresponding H_B that is convex, each H_B is clearly also a tolerance graph. Thus, for each block that is adjacent to two cut-vertices u and v on this path, the associated graph B' has a CSP that begins at a star containing u and ends at a star containing v by Lemma 2.4 as applied to H_B. For any blocks B on the ends of this path adjacent to only one cut vertex v, the same argument used in the proof of Lemma 2.4 guarantees that B' will have a CSP that begins or ends at a star containing v. Thus, we can easily combine each of these CSPs into a CSP that contains every edge in an undeleted block. Finally, each block B that was deleted from T is adjacent to a single cut-vertex v, and by Lemma 2.6 as applied to H_B, the associated graph B' has a CSP with v contained in every star. Thus we can insert this CSP into our combined CSP at the beginning or end if the first or last star already contains v, or between any two stars that both contain v. Two such stars must exist if the first and last star do not already contain v, since in this case v must be contained in two blocks that were not deleted from T. Because every edge of G is in exactly one graph B', this produces a CSP for G and so G is a tolerance graph by Theorem 1.2.

It now remains to show that the algorithm requires $O(n+m)$ steps. Finding the set P_G requires $O(n)$ time, and finding the blocks and cut-vertices of $G - P_G$ and building the block-cutpoint graph T can be done in $O(n+m)$ time [6]. Using these blocks, we can also construct the graphs $\{B' \mid B \text{ is a block of } G - P_G\}$ that partition the edges of G and the associated graphs $\{H_B \mid B \text{ is a block of } G - P_G\}$ in $O(n)$ time. The verification that each H_B is convex requires $O(n_b + m_b)$ time, where n_b and m_b are the order and size of B', respectively. Determining if $d_T(B) = 1$, and if so, identifying the cut vertex c adjacent to B in T can be done in constant time, and because B is bipartite we can determine if B' contains an induced path of length three or more that begins at v in $O(n_b)$ time. Thus, the total running time for all of these tests is $\sum_{B'} O(n_b + m_b)$

After all these tests are complete and the blocks staisfying the condition in line 8 have been deleted from T, testing that the graph remaining is a path requires $O(|V(T)|) = O(n+m)$ steps.

An easy induction proof shows that $\sum_{V(G)} b_v \leq 2n$, where b_v is the number of blocks which contain the vertex v. Hence the total running time of the algorithm is

$$\sum_{B'} O(n_b + m_b) = O(\sum_{B'} n_b + m_b) = O(m + \sum_{V(G)} b_v) = O(m+n)$$

as desired. □

Note that this algorithm also gives a new structural characterization of bipartite tolerance graphs, which we give in the following theorem:

Theorem 4.2. *Let G be a connected bipartite graph. Then G is a tolerance graph if and only if:*

(i) For every block B of G, H_B is convex

and

(ii) For each cut vertex v, G contains no induced subgraph isomorphic to the graph T_3 in Figure 1 in which v is the vertex of degree three.

As indicated in the proof of Theorem 4.1, Algorithm 1 can easily be modified to provide a CSP of G when G is a bipartite tolerance graph. This CSP can then be combined with the algorithm in [5], to give a tolerance representation of the graph G. Although this representation is not guaranteed to be polynomial in the size of G, such a polynomial sized representation is guaranteed by the result of [10]. Since the algorithm in [5] is clearly not optimal, it seems likely that there is an efficient algorithm that will convert the CSP into a polynomial sized tolerance representation of G, which could then be used to certify the correctness of the algorithm.

When Algorithm 1 returns false, we can also certify this reult, either by identifying an induced subgraph of G isomorphic to T_3, or by giving an induced subgraph H_B of G that is not convex. Although we do not have complete list of such obstructions, we can identify an asteroidal triple of H_B that is contained in V_x and an asteroidal triple of H_B contained in V_y. This certifies that H_B is not convex, and hence that G is not a tolerance graph by the contrapositive of Lemma 2.3.

Acknowledgement

The authors would like to thank R. Sritharan for his suggestions and comments.

References

[1] K. Booth and G. Lueker, Linear Algorithms to Recognize Interval Graphs and Test for the Consecutive Ones Property, in: Proceedings of the Seventh Annual ACM Symposium on Theory of Computing, 1975, 255-265.

[2] A. Brandstädt, V. B. Le, and J. P. Spinrad, Graph Classes, A Survey, Soc. for Inducstrial and Applied Math., Philidelphia, PA, 1999.

[3] A. Brandstädt, J. P. Spinrad, and L. Stewart, Bipartite permutation graphs are bipartite tolerance graphs, Congress. Numer. 58, 1987, 165-174.

[4] D. E. Brown, *Variations on Interval Graphs*, Ph.D. thesis, University of Colorado at Denver, 2004.

[5] A. H. Busch, A characterization of bipartite tolerance graphs, Discrete Applied Math. 154, 2006, 471-477.

[6] U. Derigs, O. Goecke, and R. Schrader, Bisimplicial edgs, gaussian elimination and matchings in bipartite graphs, in: Inter. Workshop on Graph-Theoretic Concepts in Compl. Sci., 1984, 79-87.

[7] M. C. Golumbic, and C. L. Monma, A generalization of interval graphs with tolerances, Congress. Numer. 35, 1982, 321-331.

[8] M. C. Golumbic, C. L. Monma, and W. T. Trotter, Tolerance Graphs, Discrete Applied Math. 9, 1984, 157-170.

[9] M. C. Golumbic, and A. N. Trenk, Tolerance Graphs, Cambridge University Press, Cambridge, UK, 2004.

[10] R. B. Hayward, and R. Shamir, A note on tolerance graph recognition, Discrete Appl. Math. 143, 2004, 307-311.

[11] L. Langley, *Interval tolerance orders and dimension*, Ph.D. thesis, Dartmouth College, 1993.

[12] H. Müller, Recognizing Interval digraphs and interval bigraphs in polynomial time, Discrete Appl. Math. 78, 1997, 189-205.

[13] L. Sheng, Cycle free probe interval graphs, Congr. Numer. 140, 1999, 33-42.

[14] J. P. Spinrad, Efficent Graph Representations. Fields Inst. Monogr. 19, Amer. Math. Soc., Providence, RI, 2003.

[15] J. P. Spinrad, A. Brandstädt, and L. Stewart, Bipartite permutation graphs, Discrete Appl. Math. 18, 1987, 279-292.

[16] A. C. Tucker, A Structure Theorem for the Consecutive 1's Property, J. Combinatorial Theory Ser. B 12, 1972, 153-162.

Appendix A: Proofs

- Proof of Observation 2.1

  Proof. Choose $u \in V(S_a) \setminus V(S_{i+1})$ for $a \leq i$ and $v \in V(S_b) \setminus V(S_i)$ for $b \geq i+1$. We will prove the stronger claim that $V(S_i) \cap V(S_{i+1})$ separates u and v. Let P be a path from u to v. Then P must contain some edge (wz) such that $w \in V(S_c)$ for $c \leq i$ and $z \in V(S_d)$ for $d \geq i+1$. Let S_t be the unique star which contains (wz). If $t \leq i$, then we have $z \in V(S_t) \cap V(S_d)$ for $t \leq i < d$, and so by the consecutive ordering of $\mathcal{S}$, $z \in V(S_i) \cap V(S_{i+1})$. Similarly, if $t \geq i+1$, $w \in V(S_i) \cap V(S_{i+1})$. Since P was chosen arbitrarily, every path from u to v contains some vertex of $V(S_i) \cap V(S_{i+1})$. □

- Proof of Observation 2.2.

  Proof. Assume $G = (V_x, V_y, E)$ is a tolerance graph. Then G has a CSP $\mathcal{S} = S_1, S_2, \ldots, S_t$. Without loss of generality, assume $c_1 \in V_x$. If $c_i \in V_x$ for each $1 \leq i \leq t$, then for any vertex $x \in V_x$, x is the center of each star it appears in, and since these stars must all be consecutive, they can combined into one star. Hence, we can assume that each vertex of V_x is in a unique star, and hence each star is the graph induced by the closed neighborhood of some vertex of V_x, and so G is X-consecutive.

  So we can assume that for some index i, $c_i \in V_y$ and let i be the minimal such index. Thus, $V_y \cap V(S_i) = \{c_i\}$ and $V_x \cap V(S_{i-1}) = \{c_{i-1}\}$ and since G is bipartite, $V(S_i) \cap V(S_{i-1}) \subseteq \{c_i, c_{i-1}\}$. Furthermore, because $\mathcal{S}$ is a partition of the edges of H_B, either $V(S_i) \cap V(S_{i+1}) \subseteq V_x$ or $V(S_i) \cap V(S_{i+1}) \subseteq V_y$. Thus, $|V(S_i) \cap V(S_{i-1})| = 1$, and since this set is a cut-set by Observation 2.1, G is not 2-connected.

  For the converse, assume that $\mathcal{N}_x$ is consecutively ordered, and let $V_x = \{x_1, v_2, \ldots x_{n_x}\}$ correspond to this ordering. Let S_i be the subgraph of G induced on $N[x_i] = N(x_i) \cup \{x_i\}$ for $1 \leq i \leq t$. Since G is bipartite, each induced subgraph is a star, and because each x_i is in a unique $V(S_i)$, the set of stars are consecutively ordered and clearly partition the edges of G. Thus, G is a tolerance graph. □

- Proof of Lemma 2.3

  Proof. Assume $G = (V_x, V_y, E)$ is a bipartite tolerance graph, and let B be a block of G. Since H_B is isomorphic to an induced subgraph of G, and tolerance graphs are hereditary, it follows that H_B is also a bipartite tolerance graph. Then by Theorem 1.2, H_B has a consecutive star partition. Let $\mathcal{S} = S_1, S_2, \ldots S_t$ be such a CSP with t maximum. If $|V(S_i) \cap V_x| = 1$ for $1 \leq i \leq t$, then each star has a central vertex in V_x, and just as in the proof of Observation 2.2, $\mathcal{N}_x$ is consecutively orderable. Similarly, if $|V(S_i) \cap V_y| = 1$ for $1 \leq i \leq t$, then $\mathcal{N}_y$ is consecutively orderable. In either case, G is convex.

So we can assume that for some indices i and j, $|V(S_i) \cap V_x| > 1$ and $|V(S_j) \cap V_y| > 1$. Without loss of generality, assume that $i < j$ and choose i and j such that $j - i$ is minimal.

We first claim that $V(S_i) \cap V(S_j) = \emptyset$. Otherwise, it must be the case that $(c_i c_j) \in E(H_B)$ and this edge is in either S_i or S_j. But we can then clearly create a CSP of H_B that has $t + 1$ stars by making this edge an additional star and inserting it immediately after S_i (if $(c_i c_j)$ is an edge of S_i) or immediately after S_j (if $(c_i c_j)$ is an edge of S_j), and removing the edge from the star in which it appears. Since this CSP has $t + 1 > t$ stars, we conclude that $V(S_i) \cap V(S_j) = \emptyset$ and since H_B is connected, we must have $j > i + 1$. Furthermore, by the minimality of $j - i$, $|V(S_m)| = 2$ for each $i < m < j$.

Next, we note that for each $i < m < j$, either S_m is a pendant edge of H_B or a cut-edge of H_B. Clearly, as H_B is connected and $\mathcal{S}$ partitions the edges of H_B, we have $|V(S_m) \cap V(S_{m\pm 1})| = 1$ for $i < m < j$. So either $V(S_{m-1}) \cap V(S_m) \cap V(S_{m+1}) = \emptyset$ and S_m is a cut edge of H_B, or $|V(S_{m-1} \cap V(S_m) \cap V(S_{m+1})| = 1$, and S_m is a pendant edge of H_B. If S_m is a pendant edge for each m, then we have $V(S_i) \cap V(S_{i+1}) \cdots V(S_{j-1}) \cap V(S_j)| = 1$. Since we showed in the previous paragraph that $V(S_i) \cap V(S_j) = \emptyset$, there must be some cut edge that separates c_i and c_j.

Recall that $|V(S_i)| \geq 3$ and $|V(S_j)| \geq 3$, and as a result c_i and c_j each have degree at least two in H_B. The only cut-edges of H_B that separate two non-pendant vertices are the edges added to B at a vertex of $\mathcal{B}^2(B)$, and as a result we can then conclude without loss of generality that S_i is a pendant path of length two, and so $i = 1$. However, in this case, we can let S_0 be the pendant edge in S_1, remove this edge from S_1 to form S'_1, and form a CSP $S_0, S'_1, \ldots, S_t$ that is longer than $\mathcal{S}$. This final contradiction establishes the lemma. □

- Proof of Lemma 2.4

Proof. Assume G is a tolerance graph, and let B be a block of G with $|\mathcal{B}^2(B)| \geq 2$. Then by the hereditary property of tolerance graphs, H_B has a consecutively ordered star partition $\mathcal{S} = S_1, S_2, \ldots, S_t$. Choose $v \in \mathcal{B}^2(B)$, and let i be the index of the star S_i that contains the edge $(v'v'')$. Note that S_i must either be the single edge $(v'v'')$ or the 2-path induced on $N[v']$. In either case, by Observation 2.1 we see that $V(S_{i\pm1}) \cap V(S_i)$ is a cut-set of H_B, and since v'' is not in any star other than S_i, and $(vv') \in E(H_B)$, we note that the set $V(S_{i\pm1}) \cap V(S_i) \subset \{v, v'\}$. Since B is 2-connected, $V(S_{i\pm1}) \cap V(S_i)$ does not separate any two vertices of B. Thus, we can conclude that $i = 1$ or $i = t$, and by deleting the vertices of $V(H_B) \setminus V(B')$, we obtain a CSP $\mathcal{S}'$ of B' with v in the first or last star.

The proof is complete by observing that for any $v_1 \neq v_2$ in $\mathcal{B}^2(B)$, the edges $v_1'v_1''$ and $v_2'v_2''$ are in distinct stars of $\mathcal{S}$, and so the CSP $\mathcal{S}'$ must begin at a star containing v_1 and end at a star containing v_2 (or vice versa), and consequently $|\mathcal{B}^2(B)| = 2$. □

- Proof of Lemma 2.6

Proof. Relabeling if necessary, we may assume $v \in V_x$. If B' does not contain any induced $2K_2$, then B' is a bipartite chain graph, and every bipartite chain graph is easily shown to be asteroidal triple free and thus also biconvex (see [15]). Therefore B' is Y-consecutive, and has a CSP with v in every star. If B' does contain edges x_1y_1 and x_2y_2 that induce a $2K_2$, then $\{v'', v', v, x_1, y_1, x_2, y_2\}$ induces a subgraph of H_B isomorphic to T_2 in Figure 1, and thus x_1, x_2, v'' form an asteroidal triple in H_B. Since H_B is convex by assumption, we conclude that H_B is Y-consecutive, and so is B'. Thus, B' has a CSP with v in every star. □