Discrete Mathematics and Theoretical Computer Science DMTCS vol. **12:5**, 2010, 63–82

Linear Time Recognition Algorithms and Structure Theorems for Bipartite Tolerance Graphs and Bipartite Probe Interval Graphs

David E. Brown[1†] and Arthur H. Busch[2‡] and Garth Isaak[3§]

[1]*Department of Mathematics and Statistics, Utah State University, Logan, UT 84341-3900*
[2]*Department of Mathematics, University of Dayton, Dayton, Ohio 45465-2316*
[3]*Department of Mathematics, Lehigh University, Bethlehem, PA 18015*

received 26th June 2009, accepted 22nd October 2010.

A graph is a *probe interval graph* if its vertices can be partitioned into probes and nonprobes with an interval associated to each vertex so that vertices are adjacent if and only if their corresponding intervals intersect and at least one of them is a probe. A graph $G = (V, E)$ is a tolerance graph if each vertex $v \in V$ can be associated to an interval I_v of the real line and a positive real number t_v such that $uv \in E$ if and only if $|I_u \cap I_v| \geq \min\{t_u, t_v\}$. In this paper we present $O(|V| + |E|)$ recognition algorithms for both bipartite probe interval graphs and bipartite tolerance graphs. We also give a new structural characterization for each class which follows from the algorithms.

Keywords: tolerance graphs, probe interval graphs, recognition algorithms

1 Introduction

We discuss undirected, finite, simple graphs G with vertex set $V(G)$ and edge set $E(G)$, and we will sometimes write $G = (V, E)$ to indicate $V = V(G)$ and $E = E(G)$. The number of vertices of a given graph G is referred to by n and the number of edges by m. If G is a bipartite graph with partite sets X and Y, we will write $G = (X, Y, E)$ to indicate this, and denote $|X|$ by n_x and $|Y|$ by n_y.

A graph G is a *probe interval graph* if there is a partition of $V(G)$ into P and N and a collection $\{I_v : v \in V(G)\}$ of closed intervals of $\mathbb{R}$ in one-to-one correspondence with $V(G)$ such that, for $u, v \in V(G)$, $uv \in E(G)$ if and only if $I_u \cap I_v \neq \emptyset$ and at least one of u, v belongs to P. The sets P and N are called the *probes* and *nonprobes*, respectively, and the collection of intervals together with such a partition will be referred to in this paper as a *probe interval representation*, but we may omit the "probe interval" if the context makes it clear.

†Email: `david.e.brown@usu.edu`.
‡Email: `art.busch@notes.udayton.edu`.
§Email: `gisaak@lehigh.edu`.

The probe interval graph model was invented in connection with the task called *physical mapping* which arose in connection with the human genome project, cf. work of Zhang and Zhang et al. [26, 27, 28].

The recognition of probe interval graphs has received considerable attention recently. If the vertex partition $V(G) = (P, N)$ into probes and nonprobes is specified as part of the input, the problem of recognizing whether G is a probe interval graph is called the *partitioned probe interval graph problem*, and this problem is solvable in time $\mathcal{O}(n^2)$ via a method involving modified PQ-trees, see [17] by Johnson and Spinrad. Another method given in [21] by McConnell and Spinrad uses modular decomposition and has complexity $\mathcal{O}(n + m \log n)$. An $O(m^2)$ algorithm for recognizing (non-partitioned) cocomparability probe interval graphs is given in [5], by Brown and Langley, which uses a modified algorithm of Golumbic and Lipshteyn [11] that recognizes graphs that are both weakly chordal[(i)] and chordal probe [(ii)]. In [9], Chang, Kloks, Liu, and Peng, present an algorithm for recognizing whether a non-partitioned graph is a probe interval graph; complexity is shown to be polynomial, but the precise order of the algorithm is not given.

A graph $G = (V, E)$ is a *tolerance graph* if to each vertex $v \in V$ there can be associated both a closed interval I_v of the real line and positive real number t_v such that $uv \in E$ if and only if $|I_u \cap I_v| \geq \min\{t_u, t_v\}$. Tolerance graphs were introduced by Golumbic and Monma in [12] as a generalization of interval graphs and to model certain scheduling problems. We will denote the collection of intervals $\{I_v : v \in V\}$ and tolerances $\{t_v : v \in V\}$ corresponding to tolerance graph $G = (V, E)$ as $\langle \mathcal{I}, t \rangle$ and refer to it as a *tolerance representation* of G, but call it simply a *representation* if the context is clear. If tolerance graph G has a representation in which $|I_v| \leq t_v$ for every $v \in V$, then G is a *bounded tolerance graph*. Additional results and background on tolerance graphs and on probe interval graphs can be found in the book by Golumbic and Trenk [14].

The recognition problem for tolerance graphs remained open for many years. In 2004, Hayward and Shamir in [15] showed that tolerance graphs always admit a polynomial sized integer representation and hence demonstrated the problem of tolerance graph recognition is in NP. Then in 2010, Mertzios, Sau and Zaks in [19] showed the the recognition problem for tolerance graphs is NP-complete. In the case of tolerance graphs which are bipartite, a characterization was given by Busch in [7] which we present here as Theorem 1.1. In [8], by Busch and Isaak, this theorem was used to give a linear time recognition algorithm for bipartite tolerance graphs. We reproduce that algorithm in Section 3, and include several lemmas that were stated without proof in [8].

The results supporting our algorithms will show subtle distinctions between the closely related classes of probe interval graphs and tolerance graphs. Initially, we note that every probe interval graph is a tolerance graph. This can be seen by the following argument: We can choose a representation for a probe interval graph G in which all interval endpoints are distinct (see, for example, Lemma 1.5 of [14]). Using the same intervals, define tolerances $t_v = \varepsilon$, if v is a probe, and $t_v = \infty$, if v is a nonprobe, we obtain a tolerance representation for G, where ε is the minimum distance between endpoints and ∞ is the length of the longest interval. The converse relationship is not true, as shown in Figure 1. One of the subtle distinctions mentioned can be seen in the two characterizations given by Theorem 1.1 and Theorem 1.2 below. A collection of sets $\mathcal{S}$ is *consecutively orderable* if the sets in the collection can be indexed $S_1, S_2, \ldots, S_t$ so that whenever $x \in S_i \cap S_k$ then $x \in S_j$ for every $i \leq j \leq k$. We call a

---

(i) A graph G is *weakly chordal* if neither G nor its complement contains an induced cycle of order $k \geq 5$.

(ii) A graph is *chordal probe* if its vertices can be partitioned into two sets P (probes) and N (nonprobes) with N an independent set and G can be extended to a chordal graph by adding edges between nonprobes.

collection of sets together with such an ordering *consecutively ordered*. When the collections of sets are the vertex sets of subgraphs of a given graph G, we will conserve notation and say that the set of subgraphs $\{G_1, G_2, \ldots, G_k\}$ is consecutively ordered when $\{V(G_1), V(G_2), \ldots, V(G_k)\}$ is. A $K_{1,n}$, for $n \geq 1$, will be called a *star* and a vertex of maximum degree will be called the *center* of the star. Obviously, when $n = 1$ the vertex identified as the center is arbitrary, but the center of the star is unique provided that $n > 1$. We next define a consecutive star partition, the key structure used in the characterization and recognition of bipartite tolerance graphs.

Definition 1.1 *Given a graph $G = (V, E)$, a set $\mathcal{G} = \{G_1, G_2, \ldots, G_t\}$ of consecutively ordered stars that partition the edges of G, will be referred to as a* consecutive star partition *(CSP) of G.*

Theorem 1.1 (Busch, [7]) *A bipartite graph G is a tolerance graph if and only if it has a consecutive star partition.*

Now, let G be a bipartite graph with vertex partition (U, N), not necessarily a bipartition, where $G(N)$ is an independent set. A *U-star* is either a star with a vertex of maximum degree in U and pendants in N (note that this permits a star with $n = 1$ as defined above), or a K_2 with both vertices in U.

Definition 1.2 *Given a graph $G = (V, E)$ and a vertex partition (U, N), a set $\mathcal{G} = \{G_1, G_2, \ldots, G_t\}$ of consecutively ordered U-stars will be referred to as a* consecutive U-star partition *(CUSP) of G.*

Theorem 1.2 (Brown, Lundgren, [6]) *A bipartite graph G is a probe interval graph if and only if it has a consecutive U-star partition.*

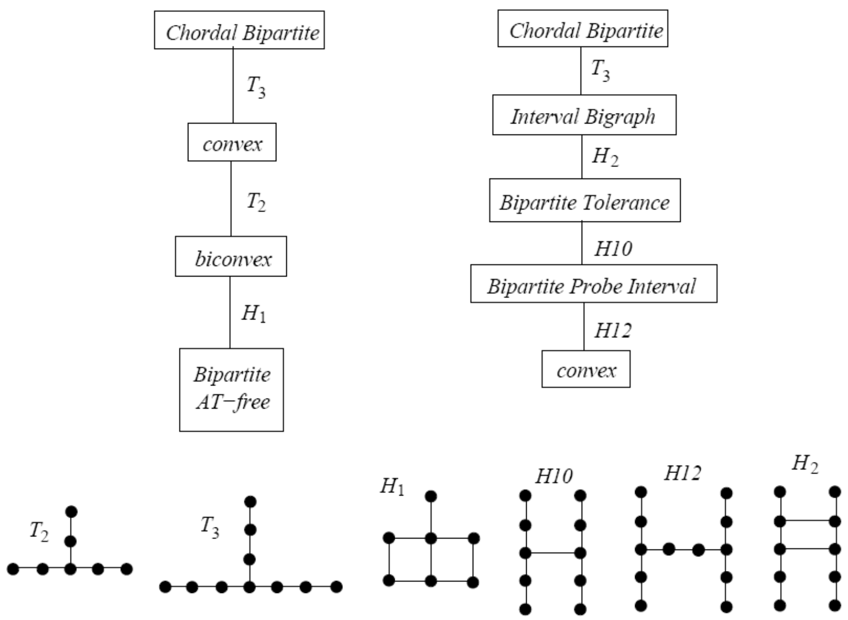

Fig. 1: Hierarchy of Chordal Bipartite graphs with separating examples.

Now, we discuss another subtle distinction between bipartite tolerance graphs and bipartite probe interval graphs. The distinction lies in the structure of blocks of a bipartite tolerance graph and in the maximal 2-edge connected subgraphs of a bipartite probe interval graph. We begin with some definitions and a result from [25] by Tucker on which we will rely.

If $G = (X, Y, E)$ is a bipartite graph with $X = \{x_1, x_2, \ldots, x_{n_x}\}$ and $Y = \{y_1, y_2, \ldots, y_{n_y}\}$, then the $n_x \times n_y$ matrix $A(G) = [m_{ij}]$ with $m_{ij} = 1$ if $x_i y_j \in E(G)$ and $m_{ij} = 0$ otherwise is the *reduced adjacency matrix of* G. $A(G)$ has the *consecutive 1's property for rows (columns)* if the columns (rows) of $A(G)$ can be permuted so that the 1's in every row (column) appear consecutively. A matrix with the consecutive 1's property for rows or for columns is called *convex* and so is the bipartite graph which corresponds to the matrix. Given a reduced adjacency matrix M for bipartite graph $G = (X, Y, E)$, an arrangement of columns so that the 1's in each row occur consecutively corresponds to an ordering of Y so that the vertices in the neighborhoods $\mathcal{N}_x = \{N(x_1), N(x_2), \ldots, N(x_{n_x})\}$ are consecutively ordered; that is, if $y_i \in N(x_r)$ and $y_k \in N(x_r)$, then $y_j \in N(x_r)$ for $i \leq j \leq k$. In this case, we say G is Y*-consecutive*; analogously, G is X*-consecutive* if the vertices in the neighborhoods $\mathcal{N}_y$ can be consecutively ordered, or equivalently, if M has the consecutive 1's property for columns. Thus, if G is X- or Y-consecutive, we say G is *convex*. If G is both X- and Y-consecutive, we say G is *biconvex*.

An *asteroidal triple* (or AT) in a graph G is a set of three vertices of G for which there is a path between any two that contains no vertex adjacent to the third.

Tucker established a nice relationship between consecutive 1's in $(0, 1)$-matrices and asteroidal triples in the corresponding bipartite graphs when the matrices are thought of as reduced adjacency matrices.

Theorem 1.3 (Tucker, [25]) *For a bipartite graph $G = (X, Y, E)$ the following are equivalent:*

1. *The reduced adjacency matrix of G has the consecutive 1's property for columns (respectively rows);*
2. *G is X-consecutive (respectively Y-consecutive);*
3. *G has no asteroidal triple contained in X (respectively no AT contained in Y).*

This result will be very useful because of what it implies about the structure of collections of consecutively orderable stars in convex graphs. Namely, for bipartite graph $G = (X, Y, E)$, if there is no asteroidal triple in X (respectively Y) then there is a collection of consecutively orderable stars which partition the edges of G all of whose centers are in X (respectively in Y). For example, if M is the reduced adjacency matrix for G and the columns of M have been permuted so that the 1s in each row are consecutive, each column j corresponds to a star S_j, and $\{S_1, S_2, \ldots, S_{n_y}\}$ is consecutively ordered.

Algorithms which determine if an $m \times n$ matrix has the consecutive 1's property for rows recognize convex graphs in $O(n{+}m)$ time. Such algorithms also form the basis of the first recognition algorithms for interval graphs, due to Booth and Lueker [1]. The algorithm of Booth and Lueker, and other algorithms to identify the consecutive 1's property, can easily be used to determine if a collection of subgraphs $\mathcal{G} = \{G_1, G_2, \ldots, G_t\}$ of G can be consecutively ordered. If $V(G) = \{v_1, v_2, \ldots, v_n\}$, simply construct the $n \times t$ *vertex-graph incidence matrix* $M = [m_{ij}]$ with $m_{ij} = 1$ if $v_i \in G_j$ 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 or U-stars which partition the edges of G, this relationship can be used to determine if G is a tolerance graph or probe interval graph using Theorem 1.3. But an arbitrary tolerance graph or probe interval graph has many partitions of its edges into stars or

U-stars, not all of which can be consecutively ordered. Therefore the above procedure cannot be used to efficiently decide if an arbitrary bipartite graph is a tolerance graph or probe interval graph.

In order to contextualize our results, Figure 1 shows some relationships between some subclasses of chordal bipartite graphs with separating examples. Recall that a graph G is *chordal bipartite* when G is bipartite and contains no induced C_k for any $k \neq 4$, and note that the class of chordal bipartite graphs properly contains the class of graphs which are both chordal and bipartite (this class is simply the forests). For other definitions, and more detail about these and other definitions, we refer the reader to [3].

The chain on the left is reproduced from [2], while the right-hand chain is due to results of Brown [4], Busch [7], Müller [20], and Sheng [22]. The classes of bounded bipartite tolerance graphs(iii), bipartite permutation graphs, unit interval bigraphs, and asteroidal-triple-free (AT-free) bipartite graphs are all equivalent, see Hell and Huang [16], and Golumbic and Trenk [14] for details about these classes of graphs and their equivalence.

The graphs T_3 and $H10$ of Figure 1 will be important for the structural description of bipartite tolerance and bipartite probe interval graphs. Their relevance is indicated by the following results of Golumbic, Monma, and Trotter, and of Sheng.

Theorem 1.4 (Golumbic, Monma, Trotter, [13]) *Let T be a tree. T is a tolerance graph if and only if T does not contain an induced subgraph isomorphic to T_3 of Figure 1.*

Theorem 1.5 (Sheng, [22]) *Let T be a tree. T is a probe interval graph if and only if T does not contain an induced subgraph isomorphic to T_3 or $H10$ of Figure 1.*

Our algorithms will be based partially on the following facts, which will be developed in Section 3 and Section 4. Within the class of 2-connected bipartite graphs, the classes of convex graphs and tolerance graphs are identical; within the class of 2-edge-connected bipartite graphs, the classes of convex graphs and probe interval graphs are identical. The former fact was observed in [8] and is this paper's Lemma 2.1; the latter is Lemma 2.2.

2 Preliminary Results for the Algorithms

We will characterize bipartite graphs whose edges can be partitioned into either sets of stars which can be consecutively ordered, or into sets of consecutively orderable U-stars, and thereby obtain conceptually simple linear time algorithms for bipartite tolerance graphs and bipartite probe interval graphs. These algorithms rest ultimately on Theorems 1.1 and 1.2, and Lemmas 2.3 and 2.4. Lemmas 2.1 and 2.1 first appeared in [8]; their proofs are included here for completeness and to emphasize the similarity between bipartite tolerance graphs and bipartite probe interval graphs.

Lemma 2.1 *If $G = (X, Y, E)$ is a connected bipartite tolerance graph with CSP $\mathcal{S} = \{S_1, S_2, \ldots, S_t\}$, then $V(S_i) \cap V(S_{i+1})$ is a cut set of G for each $i \in \{1, 2, \ldots, t-1\}$.*

Proof: Put $L = V(G) \setminus \bigcup_{j=i+1}^{t} V(S_j)$ and $R = V(G) \setminus \bigcup_{j=1}^{i} V(S_j)$. Since $\mathcal{S}$ partitions the edges of G, both L and R must be nonempty. Any edge incident with a vertex of L must be contained in a star S_j

(iii) Some authors (see [3, 10, 23]) use "bipartite tolerance graphs" instead of bounded bipartite tolerance graphs, but as indicated by Figure 1, this class is properly contained in the intersection of the classes of tolerance graphs and bipartite graphs. We have followed the convention used in [14] and used 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.

with $j \leq i$, and any edge incident with a vertex of R is contained in a star S_j with $j \geq i+1$. Thus, no edge connects L and R and $V(G) \setminus (L \cup R) = V(S_i) \cap V(S_{i+1})$ is a cut-set. □

Lemma 2.2 *If $G = (X, Y, E)$ is a connected bipartite probe interval graph with $X \cup Y$ partitioned into U and independent set N and having CUSP $\mathcal{U} = \{U_1, U_2, \ldots, U_t\}$ with respect to this partition, then each U_i that is a K_2 with both ends in U is a cut-edge of G.*

Proof: The largest induced cycle in a probe interval graph is a 4-cycle (see [18]). Note that the only way to partition a 4-cycle into consecutive stars is via two $K_{1,2}$'s. Since G is bipartite and each U_i is an induced subgraph, no U_i which is a K_2 with both vertices contained in U is part of a cycle in G. □

Proposition 2.1 *For G a 2-connected bipartite graph, G is a bipartite tolerance graph if and only if G is X-consecutive or Y-consecutive.*

Proof: Assume that $G = (X, Y, E)$ is a 2-connected tolerance graph. Then by Theorem 1.1 G has a CSP, say $\mathcal{S} = \{S_1, S_2, \ldots, S_t\}$. With c_i denoting the center of S_i, assume without loss of generality that $c_1 \in X$. If $c_i \in X$ for $1 \leq i \leq t$, then G is clearly X-consecutive. So let i be the minimal index with $c_i \in Y$. Thus, $Y \cap V(S_i) = \{c_i\}$, $X \cap V(S_{i-1}) = \{c_{i-1}\}$, and $V(S_i) \cap V(S_{i-1}) \subseteq \{c_i, c_{i-1}\}$. Since G is 2-connected, $|V(S_i) \cap V(S_{i-1})| \geq 2$ by Lemma 2.1, and so $V(S_i) \cap V(S_{i-1}) = \{c_i, c_{i-1}\}$. Since each star is an induced subgraph of G, c_i and c_{i-1} are adjacent in both S_i and S_{i-1}, contradicting the fact that $\mathcal{S}$ partitions the edges of G.

Now assume $\mathcal{N}_x$ is consecutively ordered and let $X = \{x_1, x_2, \ldots, x_{n_x}\}$ correspond to this ordering. Let S_i be the subgraph 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 S_i, the set of stars $\mathcal{S}$ is consecutively ordered and clearly partition the edges of G. Therefore, G is a tolerance graph. □

The partition of $V(G)$ into $U \cup N$, as in Theorem 1.2, results in a probe-nonprobe partition: U becomes the probe set and N becomes the nonprobe set (see Theorem 4.1 in [6]). This fact together with the next theorem gives Lemma 2.2. Recall that a bipartite graph is convex provided that it is either X-consecutive or Y-consecutive.

Theorem 2.1 (Brown, Lundgren, [6]) *Let G be a bipartite graph with bipartition $V(G) = (X, Y)$. G is a probe interval graph with probe-nonprobe partition corresponding to the given bipartition if and only if G is convex; specifically, G is X-consecutive, if and only if G has a probe-interval representation with $P = X$ (similarly for Y).*

For the recognition of bipartite probe interval graphs, we proceed in a fashion similar to that used for bipartite tolerance graphs, but instead of maximal 2-connected subgraphs, we must focus on the maximal 2-edge-connected subgraphs. If each maximal 2-edge-connected subgraph is convex then each has a CSP, following the remarks after Theorem 1.3, but we must organize the CSPs so that the centers of their stars can become the U-set as in Theorem 1.2, and so that no edge, and in particular no cut-edge of G, requires both endpoints be in N. This has implications on how the cut-edges of G are oriented with respect to G's bipartition and in which parts asteroidal triples lie. We make these relationships precise precise in Lemma 4.2. Presently, we give the analogue of Lemma 2.1 for bipartite probe interval graphs.

Proposition 2.2 *For G a 2-edge-connected bipartite graph, G is a probe interval graph if and only if G is convex.*

Proof: Let G be a 2-edge-connected bipartite probe interval graph having CUSP $\mathcal{U} = \{U_1, \ldots, U_t\}$ with respect to the partition $V(G) = U \cup N$, where $G(N)$ is an independent set. $\mathcal{U}$ has no U_i which is a K_2 contained in U, by Lemma 2.2 since G is 2-edge-connected. Thus, with c_i the center of U_i, $c_i \in U$ and $U_i \setminus \{c_i\} \subseteq N$, for each $1 \leq i \leq t$, and so no c_i is adjacent to c_j for $1 \leq i < j \leq t$. Therefore $U \cup N$ is a bipartition and G is convex by Theorem 2.1.

If $G = (X, Y, E)$ is convex with, say, $\mathcal{N}_x$ consecutively ordered, then we can take $Y = U$ and $X = N$ and the collection $\{\{y\} \cup N(y) : y \in Y\}$ is a consecutively orderable U-star partition. □

Lemmas 2.1 and 2.2 are the conceptual basis for the algorithms we present. Before presenting the algorithms, we develop some notation and identify substructures that will used by the algorithms and necessary for the statement of the structural characterizations that follow.

For the next lemmas we introduce the notation G^{k-} which stands for the graph obtained from G by iteratively deleting vertices of degree-one k times; that is, if P_G denotes the set of pendant vertices of G, then $G^{1-} = G - P_G$, $G^{2-} = G^{1-} - P_{G^{1-}}$, and so on. We will need to identify asteroidal triples contained in one or the other partite set of bipartite graph $G = (X, Y, E)$. We denote by ATX an asteroidal triple $\{x, y, z\} \subseteq X$, and by ATY, we mean an asteroidal triple $\{x, y, z\} \subseteq Y$.

Lemma 2.3 *Let G be bipartite so that G^{2-} is 2-connected. Then G is a tolerance graph if and only if G is convex.*

Proof: Every convex graph is a tolerance graph, so it suffices to show the reverse containment also holds. So assume G is a tolerance graph and G^{2-} is 2-connected, but G is not convex. By Theorem 1.1, there is a CSP $\mathcal{S} = \{S_1, S_2, \ldots, S_t\}$. If $c_i \in X$ for $1 \leq i \leq t$ or $c_i \in Y$ for $1 \leq i \leq t$, then G is convex. Otherwise, for some minimal index i, c_i and c_1 are from different partite sets of G, noting that we can take centers for S_js that are K_2s arbitrarily. Among all such CSPs of G, we choose $\mathcal{S}$ so that the index i is as large as possible and observe that this implies that $|V(S_i)| > 2$. Then by the same reasoning used in the proof of Lemma 2.1, $V(S_{i-1}) \cap V(S_i) \subset \{c_{i-1}, c_i\}$ and this implies that $c_{i-1}c_i \in E$ is a cut-edge of G. Since G^{2-} is 2-connected, the only cut-edges of G are pendant edges of G and pendant edges of G^{1-}. If $c_{i-1}c_i$ is a pendant edge, then c_{i-1} is pendant in G since $|V(S_i)| > 2$ implies that $\deg(c_i) > 1$. Thus, $E(S_{i-1}) = \{c_{i-1}c_i\}$ and $i = 2$. But by re-assigning the center of S_1 to be c_2, we obtain a CSP where S_1 and S_2 have centers in the same partite set, contradicting our choice of $\mathcal{S}$ with i maximal. Thus, we can assume that $c_{i-1}c_i$ is a pendant edge of G^{1-}, and so either c_{i-1} or c_i is pendant in G^{1-}.

If c_i is pendant in G^{1-}, then let $P = \{p_1, \ldots, p_r\}$ be the set of pendant edges of G incident with c_i. Note that the graph induced on $\bigcup_{j=1}^{i-1} V(S_j) \setminus \{c_i\}$ is the component of $G - (c_{i-1}c_i)$ which contains G^{2-}. In this case $\mathcal{S}^* = (S_1, \ldots, S_{i-2}, (S_{i-1} - c_i), (c_{i-1}c_i), p_1, p_2, \ldots, p_r)$ is a CSP of G. By assigning the center of $(c_{i-1}c_i)$ to be c_{i-1} and the center of each p_a to be the pendant vertex of G, all the centers of $\mathcal{S}^*$ are from the same partite set of G and hence G is convex.

Similarly, if c_{i-1} is pendant in G^{1-}, let $P = \{p_1, \ldots, p_r\}$ be the pendant edges of G incident with c_{i-1}. Observe that the graph induced by $\bigcup_{j=1}^{i-1} S_j$ has edge set $F \subseteq P \cup \{(c_{i-1}c_i)\}$ and since S partitions the edges of G, we conclude that $i-1 \leq |F| \leq r+1$. Furthermore, G^{2-} is contained in the component of $G-(c_{i-1}c_i)$ induced by $\bigcup_{j=i}^{t} V(S_j) \setminus \{c_{i-1}\}$ and $\mathcal{S}^* = (p_1, \ldots, p_r, (c_ic_{i-1}), (S_i - c_{i-1}), S_{i+1}, \ldots, S_t)$

is a CSP of G. By assigning the center of $(c_{i-1}c_i)$ as c_i and the center of each p_a as the pendant vertex of G for each $1 \leq a \leq r$, we note that the first star of $\mathcal{S}^*$ with central vertex in the partite set of G containing c_{i-1} has index $i^* > r + 2 \geq i$, contradicting our choice of $\mathcal{S}$. □

Lemma 2.3 along with the comments following Theorem 1.3 justify the algorithm for bipartite tolerance graphs. Briefly, if we examine the blocks of G^{2-}, assuming no T_3 of Figure 1 in G, and find each to be convex, then we can create a CSP for each block and assemble the CSPs to form one for G. Lemmas 3.1, and 3.2 of Section 3 will detail how the assembly is done.

Lemma 2.4 *Let G be a bipartite graph so that G^{2-} is 2-edge-connected. Then G is a probe interval graph if and only if G is convex.*

Proof: As every convex graph is a probe interval graph, it suffices to prove the reverse containment holds as well. So assume G is a probe interval graph and G^{2-} is 2-edge connected, but G is not convex. Then G has a CUSP $\mathcal{S} = \{S_1, S_2, \ldots, S_t\}$ by Theorem 1.2; suppose further that $\mathcal{S}$ was chosen to have the fewest possible K_2s in U and t minimal. If S_i is a K_2 in U, then by Lemma 2.2 S_i is a cut-edge, and so this edge is not an edge of G^{2-}. Now, in a manner similar to that in Lemma 2.3, via choosing i minimum, we can show that deleting this edge creates pendant edges which we can rearrange to form a new CUSP with one less K_2 in U. So by the minimality of i, no such S_i; therefore $\mathcal{S}$ is a CUSP and hence G is either X-consecutive or Y-consecutive, and hence convex. □

Lemma 2.7 forms the basis of the algorithm for bipartite probe interval graphs. Briefly, assuming bipartite graph G has no T_3 of Figure 1, we examine the maximal 2-edge-connected subgraphs of G^{2-} and check that each is convex. If each is convex, then an additional check is made to see whether the CUSPs from these subgraphs can be assembled to form a CUSP for G. Lemma 4.4 shows how the CUSPs can be assembled, and the additional check required in this algorithm is based on the structure described in Lemma 4.2.

2.1 A Structure Theorem for Bipartite Tolerance Graphs

Recall that a *block* of a connected graph G is a maximal subgraph with no cut-vertex. Clearly, every block of a connected graph is either 2-connected or a cut-edge or an isolated vertex in the trivial case where $G \cong K_1$. Let B be a block of connected graph G and define the *boundary of* B denoted ∂B as the set of vertices of B whose neighborhood is not contained in B; that is, $\partial B = \{v \in V(B) : N(v) \not\subseteq V(B)\}$. In other words, ∂B is the set of cut-vertices of G that belong to B. We partition ∂B into two sets, $\partial^1 B$, and $\partial^2 B$ as follows:

- $\partial^2 B$ contains the vertices of B in at least two blocks of G^{1-};
- $\partial^1 B = \partial B \setminus \partial^2 B$. In other words, $\partial^1 B$ consists of vertices adjacent to at least one pendant vertex and that are not in $\partial^2 B$.

For each block B of G, we define the induced subgraphs B' and B^* of G as follows:

- For each vertex u in $\partial^1 B$ include a vertex $u' \in N(u) \setminus V(B)$; the graph induced on B together with these new vertices will be denoted by B'.

- For each vertex $v \in \partial^2 B$ include vertices $v' \in N(v) \setminus V(B)$ and $v'' \in N(v') \setminus \{v\}$ in B^*. Note that because $v' \notin V(B)$, $\deg_{B^*}(v') = 2$ and $\deg_{B^*}(v'') = 1$.

See Figure 2 for an illustration. Observe that when G is 2-connected, then $\partial^2 B = \emptyset$ and $B^* = G$. More generally, Lemma 2.3 implies that if G is a bipartite tolerance graph and B is any block of G, then B^* is convex. The proof of correctness for our algorithm for bipartite tolerance graphs immediately yields the following structural result, as indicated in [8].

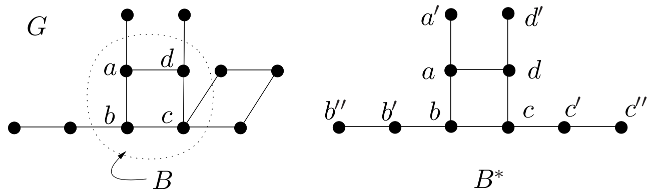

Fig. 2: A graph with block B and its corresponding B^*. $\partial^1 B = \{a, d\}$, $\partial^2 B = \{b, c\}$. We claim G is not a tolerance graph because $\{b'', a', c'\}$ and $\{d', c'', b'\}$ are asteroidal triples from different partite sets.

Theorem 2.2 *Let G be a bipartite graph. G is a tolerance graph if and only if, for any block B of G, B^* is convex and T_3 of Figure 1 is not an induced subgraph of G.*

2.2 A Structure Theorem for Bipartite Probe Interval Graphs

Let G be a connected bipartite graph and let $E^- = \{x_i y_i \mid 1 \leq i \leq t\}$ be the set of cut edges of G^{2-}. Let H be a component of $G - E^-$, and note that H may be an isolated vertex. The boundary of H, ∂H, is defined as above, and it is easy to see that $\partial H \subseteq \{x_i, y_i \mid 1 \leq i \leq t\}$. Now, for each such induced subgraph H of G, we construct the induced subgraph $\widehat{H}$ of G by including vertices $w', w'', \widehat{w} \notin V(H)$ for each $ww' \in E^-$ and $w \in \partial H$ so that the graph induced on $\{w, w', w'', \widehat{w}\}$ is a path of length three (henceforth 3-path). $\widehat{H}$ is an induced subgraph of G because, by definition, $ww' \in E^-$, and either $w'', \widehat{w}$ can be chosen from other components of $G^{2-} - E^-$, or one or both of these vertices can be chosen from $V(G) \setminus V(G^{2-})$. See Figure 3 for an illustration.

Note that when G^{2-} is 2-edge-connected, then $E^- = \emptyset$ and $\widehat{H} = G$. More generally, Lemma 2.4 implies that if G is a bipartite probe interval graph and H is any component of $G^{2-} - E^-$, then $\widehat{H}$ is convex. This fact is fundamental to our recognition algorithm for bipartite probe interval graphs. Also fundamental to the algorithm is Lemma 4.2 which we foreshadow here by way of examples. Note that $H10 = (X, Y, E)$ of Figure 1 is not convex, as it has an ATX and and ATY. Also, $H10$ has no consecutive U-star partition, but the structure it has which we generalize and use for the recognition algorithm is as follows. $(H10)^{2-}$ is a cut-edge, say xy with $x \in X$ and $y \in Y$, so the components of $(H10) - E^-$ are H_x and H_y, paths of order five centered at x and y, respectively. Since $\widehat{H}_x$ and $\widehat{H}_y$ both contain a subgraph isomorphic to T_2, we observe that $\widehat{H}_x$ contains an ATX and $\widehat{H}_y$ contains an ATY. Finally, we observe that x is in the same component of $H10 - xy$ as H_x. Our remarks in the caption of Figure 3, about $\Gamma - xy$, illustrate another example. We call this structure a *generalized* $H10$.

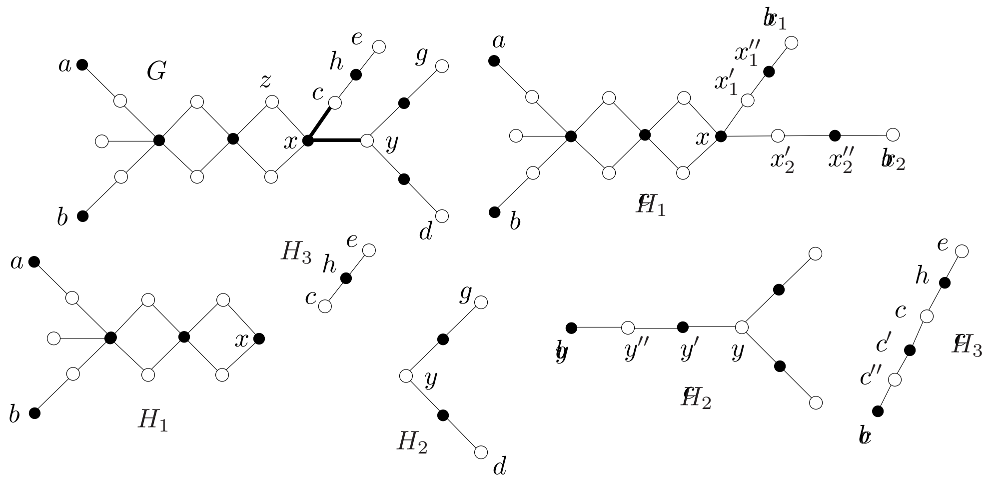

Fig. 3: A bipartite graph G with X the darkened vertices. The darkened edges cx and xy are cut-edges of G^{2-} and hence belong to E^{-}. The set $\{a, b, x\}$ is an ATX and $\{z, g, d\}$ is an ATY. G is not a probe interval graph because it has an induced T_3, but $\Gamma = G - \{c, h, e\}$ is not a probe interval graph because $\Gamma - xy$ has an ATX in $\widehat{H}_1$ and an ATY in $\widehat{H}_2$ which requires that $\{x, y\} \subset N$.

Definition 2.1 *A bipartite graph G with partition (X, Y) is a* generalized $H10$ *if G^{2-} contains 2-edge connected components H_x and H_y such that $\widehat{H}_x$ contains an ATX and $\widehat{H}_y$ contains an ATY, and every cut-edge $e = xy$ ($x \in X$, $y \in Y$) that separates H_x and H_y has the property that x and H_x are in the same component of $G - e$.*

Our recognition algorithm for bipartite probe interval graphs yields the following structural result.

Theorem 2.3 *Let G be a bipartite graph. Then G is a probe interval graph if and only if, for any component H of $G - E^{-}$, $\widehat{H}$ is convex, T_3 of Figure 1 is not an induced subgraph of G, and G does not contain a generalized $H10$.*

3 Recognizing Bipartite Tolerance Graphs

The following results describe the structure of how blocks of a bipartite tolerance graph are arranged and lead to a procedure for combining the CSPs for each B' of the graph into a CSP for the entire graph. The results of this section all appeared without proof in [8].

Lemma 3.1 *If G is a bipartite tolerance graph and B is a block of G with $|\partial^2 B| \geq 2$, then $|\partial^2 B| = 2$ and B' has a CSP with $|\partial^2 B \cap V(S_i)| = 1$ for $i = 1, t$.*

Proof: Let G be a tolerance graph and B a block of G with $|\partial^2 B| \geq 2$. Since tolerance graphs are hereditary and B^* is an induced subgraph of G, it is also a tolerance graph and thus has a CSP, say, $\mathcal{S} = (S_1, S_2, \ldots, S_t)$. Take $v \in \partial^2 B$ and let S_i be the star that contains the edge $v'v''$. Note that S_i must be the single edge $v'v''$ or the 2-path induced on $N[v']$. In either case, by Lemma 2.1, we see that $V(S_{i-1}) \cap V(S_i)$ is a cut-set of B^*, and since v'' is not in any star other than S_i, and $vv' \in E(B^*)$, we note that the set $V(S_{i-1}) \cap V(S_i)$ does not separate any two vertices of B. Thus, we conclude that $i = 1$ or $i = t$, and by deleting the vertices of $V(B^*) \setminus V(B')$, we obtain a CSP $\mathcal{S}'$ of B' with v in the first or last star.

Now observe that for any $v_1 \neq v_2$ in $\partial^2 B$, the edges $v_1'v_1''$ and $v_2'v_2''$ are in distinct stars of $\mathcal{S}$. So the CSP $\mathcal{S}'$ must begin at a star containing v_1 and end at a star containing v_2 (or vice versa). Consequently $|\partial^2 B| = 2$. □

The following corollary is a direct consequence of the contrapositive of Lemma 3.1.

Corollary 3.1 *If G is a bipartite graph and B is a block of G with $|\partial^2 B| > 2$, then B^* is not convex and G is not a tolerance graph.*

Lemma 3.2 *If $G = (X, Y, E)$ is a bipartite tolerance graph, B is a block of G with $\partial^2 B = \{v\}$, and v is at distance at most two from every other vertex of B', then B' has a CSP such that v is contained in every star.*

Proof: Let G, B, and v be as stated in the hypothesis. By Lemma 2.3, B^* is convex and so we may assume that B^* is Y-consecutive with $X = \{x_1, x_2, \ldots, x_{n_x}\}$ ordered so that $\mathcal{N}_x$ is consecutively ordered. If $v \in Y$, then this consecutive ordering induces a set of stars each of which contains v, so that by deleting vertices of $V(B^*) \setminus V(B')$, we obtain the desired CSP of B'. So, we may assume that $v \in X$. In this case the degree two vertex v' added to B' at v is in Y and that $v'' \in X$. Since v'' has degree one, it can only be adjacent to v in the consecutive ordering, and so deleting v'' and v', we obtain a consecutive ordering of $X \setminus \{v''\}$ in which v is first or last. In other words, we can assume $v'' = x_1, v = x_2$ and $N(x_2) = Y$. Now for each $y \in Y \setminus \{v'\}$, let $f(y)$ be the minimal index i such that $y \in N(x_i)$. Since $N(x_2) = Y \setminus \{v'\}$, $f(y)$ is defined for each such $y \in Y \setminus \{v'\}$ and by the consecutive ordering of $\mathcal{N}_x$, $N(y) = \{x_2, x_3, \ldots, x_{f(y)}\}$. Thus, we can index $Y \setminus \{v'\}$ so that $f(y_i) \leq f(y_j)$ whenever $i < j$ and with this ordering $(N(y_1), N(y_2), \ldots, N(y_t))$, $\mathcal{N}_y$ is consecutively ordered. The stars induced on these closed neighborhoods then form a CSP of B' with v in every star. □

Note that since tolerance graphs are hereditary, Theorem 1.4 implies that if G has an induced subgraph isomorphic to T_3, then it is not a tolerance graph.

3.1 An Algorithm for Recognizing Bipartite Tolerance Graphs

Theorem 1.1 and the results in the preceding sections provide the basis and justification for the following algorithm, presented originally in [8], which recognizes bipartite tolerance graphs in time linear with respect to the number of vertices and edges of the graph. We use $\deg_H(v)$ for the degree of vertex v in graph or subgraph H, and $\varepsilon'_H(v)$ for the maximum distance from v to any other vertex of graph or subgraph H. The *block cut-point* graph of G is the tree T with vertices the blocks and cut-vertices of G, and vertices adjacent if and only if they are incident in G.

The following proof of correctness for Algorithm 1 is nearly identical to the proof from [8]; again we include it here for comparison to Algorithm 2.

Theorem 3.1 *Algorithm 1 is correct and runs in $O(n + m)$ steps, where $n = |V(G)|$ and $m = |E(G)|$.*

Algorithm 1 G is a bipartite tolerance graph.

Input: G is a connected, bipartite graph
Output: This algorithm returns TRUE if G is a bipartite tolerance graph and returns FALSE otherwise.

Find P_G, the set of all pendant vertices of G
Find all blocks of G^{1-} and construct the block cut-point graph T of G^{1-}
for all blocks B of G^{1-} **do**
  Construct B' and B^*
  if B^* is not convex **then**
    return FALSE
  else
    if $\deg_T(B) = 1$ and $\varepsilon'_B(c) \leq 2$ 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

Fig. 4: Algorithm 1: Bipartite tolerance graph recognition algorithm.

Proof: We first show that the algorithm is correct. If $G - P_G$ contains a block B such that B^* is not convex, then G is not a tolerance graph by Lemma 2.3. If B^* is convex for every block and T is not a path, then T has a vertex of degree 3 or more. This vertex does not represent a block of $G - P_G$, since for such a block, $|\partial^2 B| \geq 3$ and B^* is not convex by Corollary 3.1. Therefore, this vertex of T represents a cut-vertex v of $G - P_G$ and is at the end of a path of length three in at least three different blocks. Thus G contains an induced subgraph isomorphic to T_3, and is not a tolerance graph by Theorem 1.4. In all other cases the algorithm returns true and we claim that a CSP for G can be constructed as follows.

For each block on the path that remains of T which 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 3.1. Similarly, for any blocks B on the ends of this path adjacent to only one cut-vertex v, B' will have a CSP that begins or ends at a star containing v. Thus, we can 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 3.2 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 at least one graph B', this produces a CSP for G and so G is a tolerance graph by Theorem 1.1. In addition, by the algorithm described in [7], a tolerance representation can be constructed from this CSP.

Now we show that the algorithm requires $O(n + m)$ steps. We use n_H and m_H to denote the number of vertices and edges respectively in a graph or subgraph H. Finding the set P_G requires $O(n)$ time, and finding the blocks and cut-vertices of $G - P_G$ and building the block cut-point graph T can be done in $O(n + m)$ time, see [24] by Tarjan. Using these blocks, we can also construct the graphs B' where B is a block of $G-P_G$ that partition the edges of G, and the adjacency matrices $A(B^*)$ can be constructed, where B is a block of $G-P_G$ in $O(n)$ time. The verification that each B^* is convex requires $O(n_{B'}+m_{B'})$ time, see [2] by Branstädt, Le, and Spinrad. Determining whether $\deg_T(B)=1$, concordantly 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 of the above tests are complete and the appropriate blocks have been deleted from T, testing that the graph which remains is a path requires $O(n_T)=O(n+m)$ steps.

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

$$\sum_{B'} O(n_{B'}+m_{B'}) = O\left(\sum_{B'}(n_{B'}+m_{B'})\right) = O\left(m+\sum_{V(G)} b_v\right) = O(n+m)$$

as desired. □

4 Recognizing Bipartite Probe Interval Graphs

In this section, we develop our algorithm which recognizes when a bipartite graph is a probe interval graph without specifying the vertex partition $V = P \cup N$. Our algorithm requires a connected bipartite graph G with partite sets X and Y labeled. We create G^{2-} and its corresponding block cut-point graph $BC(G^{2-})$. For probe interval graphs we are interested in maximal 2-edge-connected subgraphs (including isolated vertices incident only with cut edges of G^{2-}) and so we use $BC(G^{2-})$ to identify the set E^- of cut-edges of G^{2-}, and then form the bipartite graph $BC^*(G^{2-})$, whose vertices represent maximal 2-edge-connected subgraphs and cut-edges of G^{2-} and vertices are adjacent when the corresponding cut-edge and 2-edge-connected subgraph intersect. Note that T_3 of Figure 1 has the property $T_3^{2-} = BC(T_3^{2-}) = BC^*(T_3^{2-}) = K_{1,3}$. Lemma 4.1 shows that if $BC^*(G^{2-})$ is not a path, G has an induced T_3. The converse however is not true, as shown in Figure 5; $F = G - \{u, v, w\}$ has $BC^*(F^{2-})$ a path, yet F has an induced T_3.

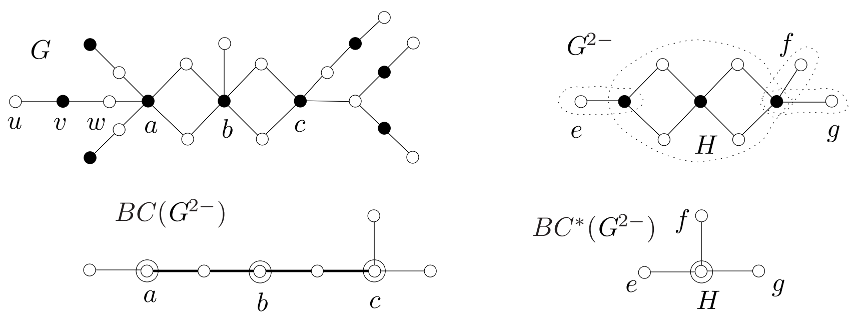

Fig. 5: A graph for which $BC^*(G^{2-})$ is not a path. Circled vertices in $BC(G^{2-})$ correspond to cut vertices; in $BC^*(G^{2-})$ they are maximal 2-edge-connected subgraphs.

Lemma 4.1 *If G is a probe interval graph and H is a component of $G - E^-$, then no more than two cut-edges of G^{2-} may be incident to H.*

Proof: Suppose H has at least three cut edges $w_1w_1', w_2w_2', w_3w_3'$ of G^{2-} incident to it with w_1, w_2, w_3 not necessarily distinct. Each edge w_iw_i' is incident to a trivial 2-edge-connected (2EC) component, the vertex w_i', or is incident to a non-trivial 2EC component. If w_iw_i' is incident to the trivial component w_i', then (1) w_i is incident to a path of length 3 $\langle w_i, w_i', w_i'', \widehat{w}_i\rangle$ with $\deg(\widehat{w}_i) = 1$, or (2) is incident to a path of length at least 2 connecting H to another non-trivial 2EC component, or (3) to a path of length at least 4 terminating at a pendant vertex. A non-trivial 2EC component contains at least an induced 4-cycle. Thus we may find an induced path of length 3 from each of w_1, w_2, w_3. So, regardless of whether w_1, w_2, w_3 are distinct, G will have an induced T_3 and cannot be probe interval. $\square$

The following corollary is a consequence of Lemma 4.1 and will justify steps 4 and 5 of the algorithm of Figure 6.

Corollary 4.1 *Suppose G is a bipartite probe interval graph and $\mathcal{H} = \{H_1, H_2, \ldots, H_{t+1}\}$ is the set of components of G^{2-} minus E^-. The elements of E^- can be indexed $e_1, e_2, \ldots, e_t$, with e_i joining H_i and H_{i+1}, for $1 \leq i \leq t$, and H_1 and H_t are incident to only one cut edge of G^{2-}.*

The contrapositive of the following Lemma gives the generalized $H10$ structure described in Definition 2.1. It will also play a key role in the proof for correctness of our probe interval graph recognition algorithm.

Lemma 4.2 *Let G be a bipartite graph with partition (X, Y), and let H_x and H_y be components of $G - E^-$ such that $\widehat{H}_x$ contains an ATX and $\widehat{H}_y$ contains an ATY. If G is a probe interval graph then there is some edge $e = xy$ that separates H_x and H_y such that y is in the same component of $G - e$ as H_x.*

Proof: Let G be a minimal counter-example. Then G is a probe interval graph which satisfies the hypothesis, but has no edge as indicated. Then G has a CUSP $\mathcal{S} = \{S_1, S_2, \ldots, S_t\}$ by Theorem 1.2. Suppose further that $\mathcal{S}$ was chosen to have the fewest possible K_2s in U and note that since G has an ATX and an ATY, there must be some S_j which is a K_2 contained in U, by Theorem 2.1. Note that by reversing our indexed order of $\mathcal{S}$ if necessary, we may assume H_x is in the graph induced by $\bigcup_{i=1}^{j-1} S_i$. We now choose the minimum index j such that S_j is a K_2 in U. If the center of S_{j-1} is in Y, then either H_y is also in the graph induced by $\bigcup_{i=1}^{j-1} S_i$ contradicting our choice of G as a minimal counter-example, or the edge S_j satisfies the lemma. On the other hand, if the center of S_{j-1} is in X, then by our choice of j, every star S_i with $i < j$ has center in X, including all the stars which include edges of H_x, contradicting Theorem 1.3. $\square$

The next result complements Lemma 2.4 and Lemma 4.1 and will justify steps 7 through 13 in Algorithm 2 of Figure 6.

Lemma 4.3 *If G is a bipartite probe interval graph, then for each maximal 2-edge-connected subgraph H of G^{2-}, $\widehat{H}$ as constructed in section 2.2 is convex.*

Proof: Assume G is a bipartite probe interval graph with bipartition (X,Y), choose a component H of $G-E^-$, and recall that $\widehat{H}$ is an induced subgraph of G; thus $\widehat{H}$ has a CUSP $\mathcal{U}=\{U_1,U_2,\ldots,U_t\}$ with respect to the partition (U,N). If either $|V(U_i)\cap X|=1$ or $|V(U_i)\cap Y|=1$, for $1\le i\le t$, then this ordering gives a consecutive ordering of the neighbors of Y or of X, respectively, and so $\widehat{H}$ is convex. So, we assume that there are indices $i<j$ for which $|V(U_i)\cap X|\ge 2$ and $|V(U_j)\cap Y|\ge 2$, relabeling partite sets if necessary. Furthermore, from the definition of a U-star, and using c_r to denote the center of U_r, it follows that $c_i\in Y$, $c_j\in X$, and that for each k satisfying $i<k<j$, U_k is a cut-edge of $\widehat{H}$. Since the only cut-edges of $\widehat{H}$ are from $E(\widehat{H})\setminus E(H)$, and each U_k separates U_i and U_j, we conclude that $i=1$, and U_1 is the path of length two induced by $\{w',w'',\widehat{w}\}$ for some $w\in\partial H\cap Y$. It then follows that $c_1=w''$ and U_2 is the edge ww'. Since U_1 is a U-star with more than two vertices, $U\cap V(U_1)=\{c_1\}$ and hence $w'\in N$, and as U_2 is also a U-star, it follows that $w\in U\cap Y$. But then either $j=3$ and $c_3=w$, contradicting our assumption that $c_3\in X$, or U_3 is a cut-edge. In the latter case, it follows immediately that $V(H)=\{w\}$. But in this case, we can conclude from Theorem 4.1 that $\widehat{H}$ is isomorphic to P_7, which is obviously convex. □

Finally, we show that the U-stars of any CUSP for any $\widehat{H}$ corresponding to bipartite probe interval graph G must be chosen so that the vertices $\widehat{w},w''$ of a 3-path incident to $w\in\partial H$ are in the first or last U-star of the CUSP. The proof of this fact is very similar to Lemma 3.1, although we note that in this case we allow for the possibility that $u=w$.

Lemma 4.4 *Suppose G is a bipartite probe interval graph and $\widehat{H}$ corresponds to some maximal 2-edge-connected subgraph of G. If the CUSP for $\widehat{H}$ is $\mathcal{U}=\{U_1,U_2,\ldots,U_t\}$ and $u,w\in\partial H$, then $\widehat{u},u''\in U_1$ and $w'',\widehat{w}\in U_t$ or vice-versa.*

Proof: With $H,u,$ and w as stated, note that uu' and ww' are cut-edges of G^{2-}. We argue for $\widehat{u},u'',u',u$ at the (wlog) beginning of the CUSP; the argument for the 3-path incident to w is similar. The edge uu' does not separate any vertices of H and so U_1 is either the K_2 on $\{\widehat{u},u''\}$ or the 2-path induced by $N[u'']$ and U_2 is either the 2-path induced on $N[u']$ or contains $\{u',u\}$, respectively. □

4.1 An Algorithm for Recognizing Bipartite Probe Interval Graphs

We are now ready to present our algorithm for recognizing bipartite probe interval graphs. For G a connected bipartite probe interval graph, note that Lemma 4.1 implies that $BC^*(G^{2-})$ has maximum degree two, and since this graph is clearly acyclic, $BC^*(G^{2-})$ must be a path. If $BC^*(G^{2-})$ is a path, each $\widehat{H}$ is checked to see that it is convex, and we use the structure of $BC^*(G^{2-})$ and a variable U*-Part* to specify whether the current $\widehat{H}_i$'s U-assignment forces the U-assignment of $\widehat{H}_{i+1}$. This U-assignment is forced if the only possible U-assignment of $\widehat{H}_i$ forces the connection point, $V(H_i)\cap e_i$, to be a non-U-vertex. Then $e_i\cap V(H_{i+1})$ must be a U-vertex. Essentially, this aspect of the algorithm tests for a generalized $H10$ in G. If the algorithm returns TRUE, then we may assemble a CUSP for G, as we show in the proof of correctness for the algorithm.

Theorem 4.1 *Algorithm 2 of Figure 6 is correct and runs in $O(n+m)$ time, where $n=|V(G)|$ and $m=|E(G)|$.*

Algorithm 2 G is a bipartite probe interval graph.

Input: Connected bipartite graph $G = (X, Y, E)$.
Output: This algorithm returns TRUE if G is a bipartite probe interval graph and returns FALSE otherwise.

Form G^{2-};
Form $BC(G^{2-})$;
Form $BC^*(G^{2-})$;
If $BC^*(G^{2-})$ is not a path **return** FALSE;
Index $E^- = \{x_i y_i \mid 1 \le i \le t\}$ the set of cut edges of G^{2-},
and $\mathcal{H} = \{H_1, \ldots, H_{t+1}\}$ the components of G– E^-, with $x_i y_i$ separating H_i and H_{i+1} ($1 \le i \le t$);
U-Part ← “”
For $i = 1$ **to** $t + 1$
  Construct $\widehat{H}_i$ from H_i
  If $\widehat{H}_i$ contains an ATX **then**
    if U-Part = “X” **then return** FALSE
      else U-Part ← “Y”
  If $\widehat{H}_i$ contains an ATY **then**
    if U-Part = “Y” **then return** FALSE
      else U-Part ← “X”
  If $i = t + 1$ **return** TRUE
  If (U-Part = “X” and $x_i \in V(H_i)$) or (U-Part = “Y” and $y_i \in V(H_i)$) **then** U-Part ← “”
end for

Fig. 6: Algorithm 2: Bipartite probe interval graph recognition algorithm.

Proof: First, we show that the algorithm is correct. If the algorithm returns FALSE, either $BC^*(G^{2-})$ is not a path, $\widehat{H}_c$ has both an ATX and an ATY for some index c, or $\widehat{H}_c$ has an ATX (or ATY) and the value of U-Part is “X” (or “Y”) for some index c. When $BC^*(G^{2-})$ is not a path, G^{2-} contains a maximal 2-edge-connected subgraph incident with three or more edges and G is not a probe interval graph by Lemma 4.1; in this case the algorithm correctly returns FALSE. If $\widehat{H}_c$ is not convex for some index c, then G is not a probe interval graph by the contrapositive of Lemma 4.3. In the last case, we claim that G contains a subgraph which is not a probe interval graph by Lemma 4.2. Now assume that the algorithm returns FALSE when $i = c$, and without loss of generality $\widehat{H}_c$ contains an ATY. Since the algorithm returns FALSE, at some previous step U-Part was set to “Y”, which happens only if $\widehat{H}_i$ contains an ATX for some $i < c$. Let a be the maximum index $a < c$ such that $\widehat{H}_a$ contains an ATX. First, note that U-Part has a value of “Y” at every step between a and c; otherwise, it must be reset to “Y” at some subsequent step prior to $i = c$, but this would imply that $\widehat{H}_b$ contains an ATX for some $a < b < c$. It now follows that $x_b \in V(H_b)$ for each index b, $a \le b < c$, otherwise $y_b \in V(H_b)$ and the algorithm would set U-Part to “”. Thus, G is not a probe interval graph by the contrapositive of Lemma 4.2: the edges of G which separate H_a and H_c are $e_a, \ldots, e_{c-1}$ and, for each separating edge $e = xy$, H_a and x are in the same component of $G - e$.

Now we claim that if the algorithm returns TRUE, a CUSP for G may be assembled and hence G is a bipartite probe interval graph by Theorem 1.2. Since $BC^*(G^{2-})$ is path, the set E^- and the components of $G - E^-$ can clearly be indexed as indicated in the algorithm. Let $X_i = V(H_i) \cap X$, $Y_i = V(H_i) \cap Y$, and re-label the vertices of $e_i = x_i y_i$ as $w_i v_i$ such that $w_i \in V(H_i)$. Since the algorithm returns TRUE, $\widehat{H}_i$ is convex for each $1 \le i \le t + 1$. Thus, by Lemma 4.4, H_1 has a CUSP $\mathcal{S}_1$ which ends at w_1,

H_{t+1} has a CUSP $\mathcal{S}_{t+1}$ which begins at v_t and H_i has a CUSP $\mathcal{S}_i$ beginning at v_{i-1} and ending at w_i for each $1 \leq i \leq t$. We will show that it is possible to construct each $\mathcal{S}_i$ with U-Part U_i such that $\mathcal{S} = (\mathcal{S}_1, e_1, \mathcal{S}_2, e_2, \ldots, \mathcal{S}_i, e_i, \ldots, \mathcal{S}_{t+1})$ is a CUSP for G with $U = \bigcup_{i=1}^{t+1} U_i$. To this end, it suffices to show that the CUSPs $\mathcal{S}_i$ can be chosen so that no e_i has both vertices assigned to N. Proceeding in a manner similar to that in Lemma 2.3, assume that among all choices of these CUSPs we have selected ones so that the minimum index c for which e_c has both its vertices in N is as large as possible and, without loss of generality, that $w_c = x_c$ and $v_c = y_c$. If $\widehat{H}_{c+1}$ contains no ATY, then this graph has a CUSP $\mathcal{S}'_{c+1}$ such that $y_c \in U_{c+1}$, and by replacing S_{c+1} with S'_{c+1} we obtain $\mathcal{S}'$ such that e_i has at least one endpoint in U for each $1 \leq i \leq c$, contradicting our choice of $\mathcal{S}_{c+1}$ so that this minimum index c was maximal. Thus, we may conclude that $\widehat{H}_{c+1}$ must have an ATY. But since the algorithm returns TRUE, we must have (1) $\widehat{H}_{c+1}$ contains no ATX, and (2) U-Part $\neq$ "Y" at the beginning of step $c+1$ of the algorithm. We can eliminate the possibility that U-Part $=$ "X" at the beginning of step $c+1$ by observing that $w_c = x_c \in V(H_c)$, so U-Part would be reset to "" by instruction 16 during step c. Thus, we can assume that U-Part $=$ "" at the end of step c and the beginning of step $c+1$. From this we conclude that $\widehat{H}_c$ contains no ATX. Furthermore, if $\widehat{H}_c$ contains an ATY, then $U_c = X_c$ and $w_c \in U$, contradicting our choice of c so that no vertex of e_c is in U. Thus, H_c is biconvex. Now, let $a < c$ be the maximum index such that $\bigcup_{i=a}^{c} V(H_i)$ is not biconvex. If no such a exists, we can choose CUSPs $\mathcal{S}'_i$ so that $U'_i = X_i$ and thus e_i has at least one vertex in U, for each $1 \leq i \leq c$ contradicting our choice of c. So a exists, and we consider the value of U-Part at the end of step a. If U-Part $=$ "", then $w_a \in U_a$ and we simply choose CUSPs $\mathcal{S}'_i$ for $a < i \leq c$ such that $U'_i = X_i$, again contradicting the choice of c so that both ends of e_c are in N. Similarly, if U-Part $=$ "X", then $w_a \notin X$ and so $v_a \in X$. Then we can choose $\mathcal{S}'_i$ for $a < i \leq c$ such that $U'_i = X_i$, and $w_c \in U$, contradicting our choice of $\mathcal{S}$ so that c was maximum. Finally, if U-Part $=$ "Y", then at some index b with $a < b \leq c$, instruction 16 must reset U-Part to "" on step b and this, in turn, implies that $w_b = y_b \in U$. In this case, we can choose CUSPs $\mathcal{S}'_i$ for $b < i \leq c$ with $U^*_i = X_i$ and obtain a CUSP $\mathcal{S}^*$ where e_i has at least one vertex in U for $1 \leq i \leq c$, contradicting our choice of $\mathcal{S}$ so that c was maximum.

We now show that the algorithm requires $O(n+m)$ steps. Clearly, constructing G^{2-} requires $O(n)$ steps, and as noted previously, constructing $BC(G^{2-})$ requires $O(n+m)$ steps, which also implies that we can determine the set E^- of cut-edges of G^{2-} in $O(n+m)$ steps. Furthermore, deleting an edge can be done in constant time, and as a result we can construct $G - E^-$ in $O(m)$ time, and determine the components of $G - E^-$ in $O(n+m)$ time. Hence constructing $BC^*(G^{2-})$, determining if this graph is a path, and if so, indexing the components of G^{2-} and edges of E^- as described in the algorithm can all be done in $O(n+m)$ steps. Now let $n_i = |V(H_i)|$ and $m_i = |E(H_i)|$. Determining ∂H_i requires at most n_i steps, and constructing $\widehat{H}_i$ from H_i will require the addition of at most 6 additional vertices and edges, since $BC^*(G^{2-})$ has been checked to be a path when $\widehat{H}_i$ is to be constructed. As in the algorithm for bipartite tolerance graphs, testing H_i for an ATX and an ATY will require $O\left((n_i+6)+(m_i+6)\right)$ steps, and determining which vertex of $e_i = x_i y_i$ is in H_i and setting the new value of U-Part can be done in constant time. Thus the total time required for the algorithm is

$$O(n+m) + \sum_{i=1}^{t+1} O\left((n_i+6)+(m_i+6)\right) = O(n+m) + O\left(\sum_{i=1}^{t+1} \left((n_i+6)+(m_i+6)\right)\right)$$

To complete the proof, we observe $\sum_{i=1}^{t+1}((n_i+6)+(m_i+6)) \leq O(n+m)$: since each vertex is in a unique H_i, $\sum_{i=1}^{t+1}(n_i+6) = n+6t+6$, and since each edge of G^{2-} is either in E^- or in a unique H_i, we have $\sum_{i=1}^{t+1}(m_i+6) \leq (m-t)+6t+6$. Finally, since t represents the number of cut edges of G^{2-}, $t = O(m)$. Therefore, Algorithm 2 runs in time $O(n+m)$. □

5 Conclusion

Algorithms 1 and 2 and the structural characterizations that result show the close relationship between convex bipartite graphs and the classes of bipartite tolerance graphs and bipartite probe interval graphs. We expect that Theorems 2.2 and 2.3 can be used to generate a list of minimal forbidden subgraphs for the classes of bipartite tolerance and bipartite probe interval graphs, respectively.

Specifically, our results show that if a bipartite graph is not a tolerance graph then it either contains a T_3 or an induced subgraph that is 2-connected after removing pendant one- and two-edge paths and is not convex. That is, it has an asteroidal triple in both sets of the bipartition. Our algorithm can be modified to output such graphs when found. Determining a list of minimal such subgraphs appears to be a bit more complicated. Tucker's list in [25] of forbidden subgraphs for an asteroidal triple in one partite set might be a starting point for developing such a list. Our results also show that if a bipartite graph is not a probe interval graph then it contains a T_3 or an induced subgraph that is two edge connected after removing pendent paths with one, two, or three edges, and is not convex or a generalized H10. Our algorithm can be modified to output such graphs when found. Developing a list of minimal forbidden subgraphs for bipartite probe interval graphs will require a list of minimal nonconvex graphs that are 2-edge-connected graphs after removing pendent paths with one, two or three edges as well as a list of minimal graphs containing a generalized H10. The blocks of these minimal graphs must have a linear structure, and, except for the end blocks, each block must be an edge, a ladder, $K_{3,3}$ minus a matching, or a 4 cycle with some additional conditions as suggested by figure 7. However details of a complete characterization of minimal generalized H10's remain incomplete.

In theory, these observations provide a foundation for a conceptually simple certification procedure when Algorithms 1 and 2 return false. However, it seems likely that the list of forbidden subgraphs for bipartite tolerance graphs and bipartite probe interval graphs will be large and include subgraphs with a large number of vertices. As a result, a more nuanced certification procedure may be desirable.

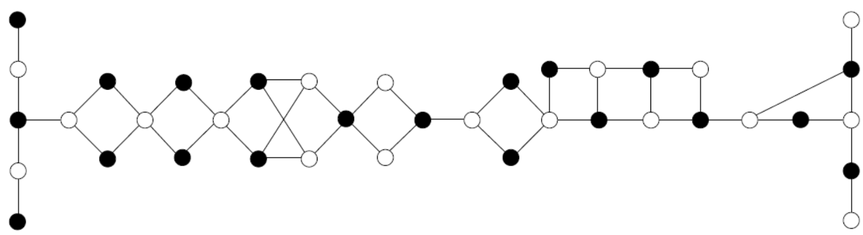

Fig. 7: A generalized H10 with the darkened vertices being the X-partition. Deleting any vertex yields a probe interval graph: removal of any pendant or cut-vertex results in a convex graph, and removal of any other vertex changes all cut-edges to those of the form described in Lemma 4.2.

References

[1] K. S. Booth and G. S. Lueker. Linear algorithms to recognize interval graphs and test for the consecutive ones property. In *Seventh Annual ACM Symposium on Theory of Computing (Albuquerque, N. M., 1975)*, pages 255–265. Assoc. Comput. Mach., New York, 1975.

[2] A. Brandstädt, Van Bang Le, and J. P. Spinrad. *Graph classes: a survey*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[3] A. Brandstädt, J. Spinrad, and L. Stewart. Bipartite permutation graphs are bipartite tolerance graphs. *Congr. Numer.*, 58:165–174, 1987. Eighteenth Southeastern International Conference on Combinatorics, Graph Theory, and Computing (Boca Raton, Fla., 1987).

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

[5] D. E. Brown and L. J. Langley. Probe interval orders. In Brams, Gehrlein, and Roberts, editors, *The Mathematics of Preference Choice and Order, Essays in Honor of P. C. Fishburn*, chapter 18, pages 313–322. Springer Berlin Heidelberg, 2009.

[6] D. E. Brown and J. R. Lundgren. Bipartite probe interval graphs, circular arc graphs, and interval point bigraphs. *Australas. J. Combin.*, 35:221–236, 2006.

[7] A. H. Busch. A characterization of triangle-free tolerance graphs. *Discrete Appl. Math.*, 154(3):471–477, 2006.

[8] A. H. Busch and G. Isaak. Recognizing bipartite tolerance graphs in linear time. In *Graph-theoretic concepts in computer science*, volume 4769 of *Lecture Notes in Comput. Sci.*, pages 12–20. Springer, Berlin, 2007.

[9] G. J. Chang, A. J. J. Kloks, J. Liu, and S.-L. Peng. The PIGs Full Monty—a floor show of minimal separators. In *STACS 2005*, volume 3404 of *Lecture Notes in Comput. Sci.*, pages 521–532. Springer, Berlin, 2005.

[10] O. Goecke Dergis, U. and R. Schrader. Bisimplicial edges, gaussian elimination and matchings in bipartite graphs. In *Workshop in Graph Theoretic Concepts in Comp. Sci.*, pages 79 –87, 1984.

[11] M. C. Golumbic and M. Lipshteyn. Chordal probe graphs. *Discrete Appl. Math.*, 143(1-3):221–237, 2004.

[12] M. C. Golumbic and C. L. Monma. A generalization of interval graphs with tolerances. In *Proceedings of the thirteenth Southeastern conference on combinatorics, graph theory and computing (Boca Raton, Fla., 1982)*, volume 35, pages 321–331, 1982.

[13] M. C. Golumbic, C. L. Monma, and W. T. Trotter, Jr. Tolerance graphs. *Discrete Appl. Math.*, 9(2):157–170, 1984.

[14] M. C. Golumbic and A. N. Trenk. *Tolerance graphs*, volume 89 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2004.

[15] R. B. Hayward and R. Shamir. A note on tolerance graph recognition. *Discrete Appl. Math.*, 143(1-3):307–311, 2004.

[16] P. Hell and J. Huang. Interval bigraphs and circular arc graphs. *J. Graph Theory*, 46(4):313–327, 2004.

[17] J. L. Johnson and J. P. Spinrad. A polynomial time recognition algorithm for probe interval graphs. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, DC, 2001)*, pages 477–486, Philadelphia, PA, 2001. SIAM.

[18] F. R. McMorris, C. Wang, and P. Zhang. On probe interval graphs. *Discrete Appl. Math.*, 88(1-3):315–324, 1998.

[19] G. B. Mertzios, I. Sau, and S. Zaks. The recognition of tolerance and bounded tolerance graphs. In *Symposium on Theoretical Aspects of Computer Science*, pages 585–596, 2010.

[20] H. Müller. Recognizing interval digraphs and interval bigraphs in polynomial time. *Discrete Appl. Math.*, 78(1-3):189–205, 1997.

[21] McConnell R.M. and J.P. Spinrad. Construction of probe interval models. In: Proc. 13th ACM-SIAM Symp. on Discrete Algs. (SODA02), pages 866–875. Association for Computing Machinery, San Francisco, CA, 2002.

[22] L. Sheng. Cycle free probe interval graphs. In *Proceedings of the Thirtieth Southeastern International Conference on Combinatorics, Graph Theory, and Computing (Boca Raton, FL, 1999)*, volume 140, pages 33–42, 1999.

[23] J. P. Spinrad, A. Brandstädt, and L. Stewart. Bipartite permutation graphs. *Discrete Appl. Math.*, 18(3):279–292, 1987.

[24] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[25] A. Tucker. A structure theorem for the consecutive 1's property. *J. Combinatorial Theory Ser. B*, 12:153–162, 1972.

[26] P. Zhang. Probe interval graphs and its applications to physical mapping of DNA. *manuscript*, 1994.

[27] P. Zhang. Methods of mapping DNA fragments. United States Patent. http://www.cc.columbia.edu/cu/cie/techlists/patents/5667970.htm, 1997.

[28] P. Zhang, E.A. Schon, S.F. Fischer, E. Cayanis, J. Weiss, S. Kistler, and P.E. Bourne. An algorithm based on graph theory for the assembly of contigs in physical mapping of DNA. *CABIOS*, 10:309–317, 1994.