Contributions to Discrete Mathematics

Volume 19, Number 3, Pages 102–120
ISSN 1715-0868

ON CHARACTERIZATION AND RECOGNITION OF PROPER TAGGED PROBE INTERVAL GRAPHS

SANCHITA PAUL, SHAMIK GHOSH, SOURAV CHAKRABORTY, AND MALAY SEN

Abstract. Interval graphs were used in the study of the human genome project by the molecular biologist Benzer. Later on probe interval graphs were introduced by Zhang as a generalization of interval graphs for the study of cosmid contig mapping of DNA. Further research in this area required more useful and cost-effective tools. The concept of tagged probe interval graphs is motivated from this point of view. In this paper, we consider a natural subclass of it, namely, the class of proper tagged probe interval graphs. In this paper, we present a characterization theorem and a linear time recognition algorithm for proper tagged probe interval graphs. Also, we discuss the interrelations between the classes of proper tagged probe interval graphs and tagged probe interval graphs with probe interval graphs and probe proper interval graphs.

1. Introduction

One of the most intriguing problems in molecular biology, especially in the human genome project, is the physical mapping of DNA that aims to reconstruct the relative position of DNA fragments along the genome. In 1959, Benzer [2] applied interval graphs to obtain such a physical map from information on pairwise overlaps of the fragments. A graph $G = (V, E)$ is an *interval graph* if one can map each vertex into an interval on the real line so that any two vertices are adjacent if and only if their corresponding intervals intersect. A natural and well-studied subclass of interval graphs is

Received by the editors June 4, 2019, and in revised form July 4, 2022.

2000 *Mathematics Subject Classification.* Primary 05C62, 05C75, 05C85, Secondary 05C50.

Key words and phrases. Interval graph, proper interval graph, probe interval graph, probe proper interval graph, tagged probe interval graph, consecutive 1's property, PQ-tree algorithm.

A preliminary version of this article appeared in *Characterization and Recognition of Proper Tagged Probe Interval Graphs*, Intelligent Computing - Proceedings of the Computing Conference 2019, pp 62–75, Advances in Intelligent Systems and Computing, vol 998. Springer, Cham. This research is supported by the UGC (University Grants Commission) NET fellowship (21/12/2014(ii)EU-V) of the first author.

the class of proper interval graphs. A *proper interval graph* G is an interval graph with an interval representation of G such that no interval contains another properly. There are several characterizations of proper interval graphs [5, 6, 8, 11].

In 1994, Zhang introduced the concept of probe interval graphs which worked successfully as a model for a new concept known as cosmid contig mapping [10, 17]. This generates overlapped information by hybridization. A set of clones is placed on a filter for colony hybridization, and the filter is probed with labeled clones. Thus the set of vertices (clones) is partitioned into probes and nonprobes, and the adjacency requires the overlap information between a pair of clones only when one of them is a probe. A graph $G = (V, E)$ is a *probe interval graph* (PIG) if the vertex set V is partitioned into two disjoint sets, probe vertices P and nonprobe vertices N and one can map each vertex $x \in V$ into an interval I_x on the real line such that there is an edge between two vertices x and y if and only if at least one of them is in P and $I_x \cap I_y \neq \emptyset$. In 2010, Ghosh, Podder, and Sen [7] obtained a characterization of probe interval graphs in terms of the adjacency matrix. When the interval representation of PIG is proper (i.e., no interval is properly contained in another interval), the graph is called a *probe proper interval graph* (PPIG). In 2014, Nussbaum [11] gave a linear time algorithm for probe proper interval graphs.

Shortly after the introduction of probe interval graphs, the development of research in molecular biology requires further refinements. In the new model, a set of clones (probes) are radioactively labeled at their ends, and one can easily detect the overlapping of a pair of clones when one is labeled and the other contains at least one end of the labeled one. To capture this model, the concept of tagged probe interval graphs is defined in [14, 15, 16]. A graph $G = (V, E)$ is a *tagged probe interval graph* (TPIG) if the vertex set V can be partitioned into two disjoint sets P (called "probe vertices") and N (called "nonprobe vertices") and one can map each vertex into an interval on the real line (vertex $x \in V$ mapped to $I_x = [l_x, r_x]$) such that N is an independent set in G, there is an edge between $x, y \in P$ if and only if $I_x \cap I_y \neq \emptyset$ and there is an edge between $x \in P$ and $y \in N$ if and only if either $l_x \in I_y$ or $r_x \in I_y$. We call the collection $\{I_x \,|\, x \in V\}$ a TPIG *representation* of G. If the partition of the vertex set V into probe and nonprobe vertices is given, then we denote the graph as $G = (P, N, E)$.

Interestingly, when the subgraph of a TPIG induced by its probe vertices is proper, then each pair of distinct intervals corresponding to probe vertices contain one endpoint of the other, and so it helps the overlap detection process. Also, in some biological frameworks, the set of clones is virtually inclusion-free, especially when all clones have similar lengths as in the case of cosmid clones. In this case, the physical mapping problem can be modeled using proper interval structures [9]. Thus it becomes interesting to consider the subclass of the class of tagged probe interval graphs, namely, proper tagged probe interval graphs. A tagged probe interval graph

$G = (P, N, E)$ is a *proper tagged probe interval graph* (PTPIG) if G has a TPIG representation $\{I_x \mid x \in P \cup N\}$ such that $\{I_p \mid p \in P\}$ is a proper interval representation of G_P where G_P is the subgraph induced by the vertex set P. We call such an interval representation a *PTPIG representation* of G.

Since the inception of tagged probe interval graphs, there is still no characterization theorem or recognition algorithm for this class of graphs or any natural subclass of it, except probe proper interval graphs. However, it is proved that, like probe interval graphs, tagged probe interval graphs are also weakly chordal and hence perfect [14]. In this paper, we obtain a characterization theorem and a linear time recognition algorithm for proper tagged probe interval graphs. We hope these results will benefit molecular biologists involved in the human genome project. For more detailed information on applications of interval graphs, probe interval graphs, and tagged probe interval graphs in molecular biology and other areas, one may consult [4, 8, 14, 17].

2. Preliminaries

Let $G = (V, E)$ be a graph and $v \in V$. Then the *closed neighborhood* of v in G is the set $N[v] = \{u \in V \mid u \text{ is adjacent to } v\} \cup \{v\}$. A graph is called *reduced* if no two vertices have the same closed neighborhood. If the graph is not reduced then we define an equivalence relation on the vertex set V such that v_i and v_j are equivalent if and only if v_i and v_j have the same (closed) neighbors in V. Each equivalence class under this relation is called a *block* of G. For any vertex $v \in V$ we denote the equivalence class containing v by $B(v)$. The *reduced graph* of G (denoted by $\widetilde{G} = (\widetilde{V}, \widetilde{E})$) is the graph obtained by merging all the vertices that are in the same equivalence class. A *straight enumeration* of G is a linear ordering of blocks in G, such that for every block, the block and its neighbouring blocks are consecutive in the ordering.

If M is a $(0, 1)$-matrix, then we say M satisfies the *consecutive* 1*'s property* if in each row and column, 1's appear consecutively. We will denote by $A(G)$ the *augmented adjacency matrix* of the graph G, in which all the diagonal entries are 1, and nondiagonal elements are the same as the adjacency matrix of G.

The class of proper interval graphs is an extremely rich class of graphs with several other characterizations of it. Among them, we repeatedly use the following equivalent conditions in the rest of the paper:

Theorem 2.1 ([5, 6, 8, 11])**.** *Let $G = (V, E)$ be an interval graph. then the following are equivalent:*

1. *G is a proper interval graph.*
2. *There is an ordering of V such that for all $v \in V$, elements of $N[v]$ are consecutive (the closed neighborhood condition).*

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	1	1	0	0	0	0	0	0
v_2	1	1	1	1	1	0	0	0
v_3	0	1	1	1	1	1	0	0
v_4	0	1	1	1	1	1	0	0
v_5	0	1	1	1	1	1	1	0
v_6	0	0	1	1	1	1	1	0
v_7	0	0	0	0	1	1	1	1
v_8	0	0	0	0	0	0	1	1

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	
v_1	1	2	1						
v_2			3	4	5	2			
v_3						6	3		
v_4							4		
v_5							7	5	
v_6								6	
v_7								8	7
v_8									8

TABLE 1. The matrix $A(G)$ with its stair partition and the canonical sequence (1 2 1 3 4 5 2 6 3 4 7 5 6 8 7 8) of $A(G)$ of the graph G in Example 3.2.

(3) *There is an ordering of V such that the augmented adjacency matrix $A(G)$ of G satisfies the consecutive 1's property.*

(4) *There is an ordering $\{v_1, v_2, \ldots, v_n\}$ of V such that G has a proper interval representation $\{I_{v_i} = [a_i, b_i] \mid i = 1, 2, \ldots, n\}$ where $a_i \neq b_j$, $i, j \in \{1, 2, \ldots, n\}$ and $a_1 < a_2 < \cdots < a_n$ and $b_1 < b_2 < \cdots < b_n$.*

(5) *G has a straight enumeration which is unique up to reversal, if G is connected.*

Remark 2.2: We note that in a proper interval graph $G = (V, E)$, the ordering of V that satisfies any one of the conditions (2), (3) and (4) in the above theorem also satisfies the other conditions among them. Henceforth we call such an ordering, a *natural* or *canonical* ordering of V. But this canonical ordering is not unique. Interestingly, it follows from Corollary 2.5 of [6] (also see [11]) that the canonical ordering is unique up to reversal for a connected reduced proper interval graph.

3. Canonical Sequence of Proper Interval Graphs

Definition 3.1. *Let $G = (V, E)$ be a proper interval graph with $V = \{v_i \mid i = 1, 2, \ldots, n\}$ and $A(G)$ be the augmented adjacency matrix of G with consecutive 1's property. We partition positions of $A(G)$ into two sets (L, U) by drawing a polygonal path from the upper left corner to the lower right corner such that the set L [resp. U] is closed under leftward or downward [respectively, rightward or upward] movement (called a* stair partition *[1]) and U contains precisely all the zeros right to the principal diagonal of $A(G)$ (see Table 1(left)). We obtain a sequence of positive integers belonging to $\{1, 2, \ldots, n\}$, each occurs exactly twice, by writing the row or column numbers as they appear along the stair. We call this sequence, the* canonical sequence *of $A(G)$ (see Table 1(right)).*

There is an alternative way to get this sequence from the interval representation of G. Let $\{v_1, v_2, \ldots, v_n\}$ be a canonical ordering of the set V with the interval representation be $\{I_{v_i} = [a_i, b_i] \mid i = 1, 2, \ldots, n\}$ where $a_i \neq b_j$ for all $i, j \in \{1, 2, \ldots, n\}$, $a_1 < a_2 < \cdots < a_n$ and $b_1 < b_2 < \cdots < b_n$. We combine all a_i and b_i $(i = 1, 2, \ldots, n)$ in a single increasing sequence (say $\mathcal{I}_G$, the *interval canonical sequence*) and replace a_i or b_i by i for all $i = 1, 2, \ldots, n$, then we obtain a sequence of integers belonging to $\{1, 2, \ldots, n\}$ each occurring twice. We denote this sequence by $\mathcal{S}_G$. Moreover if we replace i by v_i for all $i = 1, 2, \ldots, n$ in $\mathcal{S}_G$ we get a sequence of vertices (say, $\mathcal{V}_{G_P}$, the *vertex canonical sequence*) of G. A similar concept described in a different language is found in [13].

Example 3.2. *Consider the proper interval graph $G_P = (P, E)$ where $P = \{p_i | i = 1, 2, \ldots, 8\}$ in Example 4.3. Let $[a_i, b_i]$ be the interval corresponding to the vertex p_i for $i = 1, 2, \ldots, 8$ in a proper interval representation of G_P,where*

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
1	2	4	5	6	8	11	14	3	7	9	10	12	13	15	16

Then the sequence combining a_i and b_i is given by

$$\mathcal{I}_{G_P} = (a_1, a_2, b_1, a_3, a_4, a_5, b_2, a_6, b_3, b_4, a_7, b_5, b_6, a_8, b_7, b_8).$$

Therefore $\mathcal{S}_G = (1\ 2\ 1\ 3\ 4\ 5\ 2\ 6\ 3\ 4\ 7\ 5\ 6\ 8\ 7\ 8)$ and $\mathcal{V}_G = (v_1, v_2, v_1, v_3, v_4, v_5, v_2, v_6, v_3, v_4, v_7, v_5, v_6, v_8, v_7, v_8)$ with respect to the given canonical vertex ordering.

Lemma 3.3. *Let $G = (V, E)$ be a proper interval graph and $\{v_1, v_2, \ldots, v_n\}$ be a canonical ordering of vertices of G. Then the sequence $\mathcal{S}_G$ is independent of proper interval representations that satisfy the given canonical ordering. Moreover, $\mathcal{S}_G$ is unique up to reversal for connected reduced proper interval graphs.*

Proof. Let $\{I_{v_i} = [a_i, b_i] \mid i = 1, 2, \ldots, n\}$ and $\{J_{v_i} = [c_i, d_i] \mid i = 1, 2, \ldots, n\}$ be two proper interval representations of G that satisfy the given canonical ordering. We have for any $i < j$, $a_j < b_i$ if and only if v_i is adjacent to v_j if and only if $c_j < d_i$. Thus the sequence $\mathcal{S}_G$ is independent of proper interval graph representations. Since the canonical ordering is unique up to reversal for a connected reduced proper interval graph, the sequence $\mathcal{S}_G$ is unique up to reversal for connected reduced proper interval graphs. □

In the following, we will show that the canonical sequence of $A(G)$ is the same as $\mathcal{S}_G$ with respect to the given canonical ordering of vertices of G. Moreover, $\mathcal{S}_G$ and its corresponding $\mathcal{V}_G$ and $\mathcal{I}_G$ can be obtained uniquely from each other. Hence abuse of notations, we will use the term canonical sequence to mean any of these throughout the paper.

Theorem 3.4. *Let $G = (V, E)$ be a proper interval graph with a canonical ordering $V = \{v_1, v_2, \ldots, v_n\}$ of vertices of G. Let $A(G)$ be the augmented* *adjacency matrix of G arranging vertices in the same order as in the canonical ordering. Then the sequence $\mathcal{S}_G$ of G is the same as the canonical sequence of $A(G)$.*

Proof. We show that the canonical sequence of $A(G)$ is an $\mathcal{S}(G)$ sequence for some proper interval representation. So the proof follows by the uniqueness mentioned in Lemma 3.3. Given the matrix $A(G)$, a proper interval representation of G is obtained as follows. Let $a_i = i$ and $b_i = U(i) + 1 - \frac{1}{i}$ where $U(i) = \max\{j \mid j \geq i \text{ and } v_iv_j = 1 \text{ in } A(G)\}$ for each $i = 1, 2, \ldots, n$. Then $\{I_{v_i} = [a_i, b_i] \mid i = 1, 2, \ldots, n\}$ is a proper interval representation of G [6]. If $U(1) > 1$, to make all the endpoints distinct, we slightly increase the value of b_1 (which is the only integer-valued right endpoint and is equal to $a_{U(1)}$) so that it is still less than its nearest endpoint which is greater than it. Thus we get a proper interval representation of G that satisfies the condition 4 of Theorem 2.1. Then the sequence $\mathcal{S}_G$ merges with the canonical sequence of $A(G)$ for this proper interval representation of G as for $i < j$, $a_j = j < b_i$ if and only if $v_iv_j = 1$ if and only if the column number j appears before the row number i in the canonical sequence of $A(G)$. □

Remark 3.5: We note that it follows from the above theorem and Lemma 1 of [13] or Corollary 2.5 of [6] that for any connected proper interval graph G, $\mathcal{S}_G$ is unique up to reversal.

4. Structure of PTPIG

Let us consider a graph $G = (V, E)$, in general, with an independent set N and $P = V \setminus N$ such that the subgraph G_P of G induced by P is a proper interval graph. Let us order the vertices of P in a canonical ordering. The adjacency matrix of G looks like the following:

	P	N
P	$A(P)$	$A(P, N)$
N	$A(P, N)^T$	$\mathbf{0}$

Note that the (augmented) adjacency matrix $A(P)$ of G_P satisfies the consecutive 1's property, and the $P \times N$ submatrix $A(P, N)$ of the adjacency matrix of G represents edges between probe vertices and nonprobe vertices. For convenience, henceforth, a continuous stretch (a subsequence of consecutive entries) in a canonical sequence will be called a *substring*.

Theorem 4.1. *Let $G = (V, E)$ be a graph with an independent set N and $P = V \setminus N$ such that G_P, the subgraph induced by P is a proper interval graph. Then G is a proper tagged probe interval graph with probes P and nonprobes N if and only if there is a canonical ordering of vertices belonging to P such that the following condition holds:*

(A) *For every nonprobe vertex $w \in N$, there is a substring in the canonical sequence with respect to the canonical ordering such that all the* *vertices in the substring are neighbors of w and all the neighbors of w are present at least once in the substring.*

Proof. Necessary condition: Let $G = (V, E)$ be a PTPIG with probes P and nonprobes N such that $V = P \cup N$. Let $\{I_x = [\ell_x, r_x] \mid x \in V\}$ be a PTPIG representation of G such that $\{I_u \mid u \in P\}$ be a proper interval representation of G_P. Then a probe vertex $u \in P$ is adjacent to $w \in N$ if and only if $\ell_u \in I_w$ or $r_u \in I_w$. Let $u_1, u_2, \ldots, u_p$ be a canonical ordering of vertices in P that satisfies the conditions of Theorem 2.1. Consider the corresponding canonical sequence $\mathcal{S}_{G_P}$ which is obtained from the combined increasing sequence of ℓ_{u_i} and r_{u_i} for $i = 1, 2, \ldots, p$. Since both sequences ℓ_{u_i} and r_{u_i} are increasing and I_w is an interval, all the ℓ_{u_i}'s and r_{u_i}'s which are belonging to I_w occur consecutively in the canonical sequence. Thus for any $w \in N$ there exists a substring of $\mathcal{S}_{G_P}$ such that all the vertices in the substring are neighbors of w and all the neighbors of w are present at least once in the substring.

Sufficiency condition: Let $G = (V, E)$ be a graph with an independent set N and $P = V \setminus N$ such that G_P, the subgraph induced by P is a proper interval graph, $P = \{u_1, u_2, \ldots, u_p\}$ and $N = \{w_1, w_2, \ldots, w_q\}$. Suppose there is a canonical ordering $u_1, u_2, \ldots, u_p$ of vertices belonging to P such that for any nonprobe vertex $w \in N$, there is a substring in the canonical sequence $S = \mathcal{S}_{G_P}$ with respect to this canonical ordering such that all the vertices in the substring are neighbors of w and all the neighbors of w are present at least once in the substring. Let us count the positions of each element in S from 1 to $2p$. For each probe vertex u_i, we assign the closed interval $[\ell_{u_i}, r_{u_i}]$ such that ℓ_{u_i} and r_{u_i} are position numbers of first and second occurrences of i in S respectively. By definition of a canonical sequence, we have $\ell_{u_1} < \ell_{u_2} < \cdots < \ell_{u_p}$ and $r_{u_1} < r_{u_2} < \cdots < r_{u_p}$. Also since all position numbers are distinct, $\ell_{u_i} \neq r_{u_j}$ for all $i, j \in \{1, 2, \ldots, p\}$. Thus this interval representation obeys the given canonical ordering of vertices belonging to P and by construction, the canonical sequence with respect to it is same as S.

We show that this interval representation is indeed an interval representation of G_P which is proper. Let $i < j$, $i, j \in \{1, 2, \ldots, p\}$. Then $\ell_{u_i} < \ell_{u_j}$ and $r_{u_i} < r_{u_j}$. Thus none of $[\ell_{u_i}, r_{u_i}]$ and $[\ell_{u_j}, r_{u_j}]$ contains other properly. We have u_i is adjacent to u_j in G_P if and only if $u_i u_j = 1$ in $A(P)$ when vertices of $A(P)$ are arranged as in the given canonical ordering. Again $u_i u_j = 1$ with $i < j$ if and only if j is lying between two occurrences of i in the canonical sequence of $A(P)$ and hence in S by Theorem 3.4. Also since $i < j$, the second occurrence of j is always after the second occurrence of i in S. Thus $u_i u_j = 1$ with $i < j$ if and only if $\ell_{u_j} \in [\ell_{u_i}, r_{u_i}]$. This completes the verification that $\{[\ell_{u_i}, r_{u_i}] \mid i = 1, 2, \ldots, p\}$ is a proper interval representation of G_P and that corresponds to S.

Next, for each $j = 1, 2, \ldots, q$, consider the substring in the canonical sequence S such that all the vertices in the substring are neighbors of w_j and all the neighbors of w_j are present at least once in the substring. Let the substring start at ℓ_{w_j} and end at r_{w_j} in S. Then we assign the interval $[\ell_{w_j}, r_{w_j}]$ to the vertex w_j. If w_j is an isolated vertex, then we assign a closed interval whose endpoints are greater than ℓ_{u_i} and r_{u_i} for all $i = 1, 2, \ldots, p$. It suffices to show that $\{[\ell_{u_i}, r_{u_i}] \mid i = 1, 2, \ldots, p\} \cup \{[\ell_{w_j}, r_{w_j}] \mid j = 1, 2, \ldots, q\}$ is a PTPIG representation of G, i.e., if u_i is a probe vertex and w_j is a nonprobe vertex then there is an edge between them if and only if either $\ell_{u_i} \in [\ell_{w_j}, r_{w_j}]$ or $r_{u_i} \in [\ell_{w_j}, r_{w_j}]$. Note that since N is an independent set by definition, there are no edges between nonprobe vertices.

First let us assume that there is an edge between u_i and w_j. Therefore the vertex u_i must be present in the substring of S that contains all the neighbors of w_j and contains only the neighbors of w_j. Since ℓ_{w_j} and r_{w_j} are the beginning and ending positions of the substring respectively, either ℓ_{u_i} or r_{u_i} must be in the interval $[\ell_{w_j}, r_{w_j}]$. Conversely, let either $\ell_{u_i} \in [\ell_{w_j}, r_{w_j}]$ or $r_{u_i} \in [\ell_{w_j}, r_{w_j}]$. Then we have either ℓ_{u_i} or r_{u_i} must be present in the substring. Since the substring contains vertices that are neighbors of w_j, we have u_i must be a neighbor of w_j. □

Remark 4.2: If G is a PTPIG such that G_P is connected and reduced, then there is a unique (up to reversal) canonical ordering of vertices belonging to P, as we mentioned in Remark 2.2. Thus the corresponding canonical sequence is also unique up to reversal. Also if condition (A) holds for a canonical sequence, it also holds for its reversal. Thus in this case condition (A) holds for **any** canonical ordering of vertices belonging to P.

		$[1,3]$ p_1	$[2,7]$ p_2	$[4,9]$ p_3	$[5,10]$ p_4	$[6,12]$ p_5	$[8,13]$ p_6	$[11,15]$ p_7	$[14,16]$ p_8
$[1,3]$	p_1	1	1	0	0	0	0	0	0
$[2,7]$	p_2	1	1	1	1	1	0	0	0
$[4,9]$	p_3	0	1	1	1	1	1	0	0
$[5,10]$	p_4	0	1	1	1	1	1	0	0
$[6,12]$	p_5	0	1	1	1	1	1	1	0
$[8,13]$	p_6	0	0	1	1	1	1	1	0
$[11,15]$	p_7	0	0	0	0	1	1	1	1
$[14,16]$	p_8	0	0	0	0	0	0	1	1

		$[6,8]$ n_1	$[4,10]$ n_2	$[10,16]$ n_3	$[1,10]$ n_4	$[17,17]$ n_5	$[1,16]$ n_6
$[1,3]$	p_1	0	0	0	1	0	1
$[2,7]$	p_2	1	1	0	1	0	1
$[4,9]$	p_3	0	1	0	1	0	1
$[5,10]$	p_4	0	1	1	1	0	1
$[6,12]$	p_5	1	1	1	1	0	1
$[8,13]$	p_6	1	1	1	1	0	1
$[11,15]$	p_7	0	0	1	0	0	1
$[14,16]$	p_8	0	0	1	0	0	1

TABLE 2. A proper tagged probe interval representation of the graph G in Example 4.3.

Example 4.3. *Consider the graph $G = (V, E)$ with an independent set $N = \{n_1, n_2, \ldots, n_6\}$ and $P = V \setminus N = \{p_1, p_2, \ldots, p_8\}$, where the matrices $A(P)$ and $A(P, N)$ are given in Table 2. First note that $A(P)$ satisfies consecutive 1's property. Hence G_P is a proper interval graph. Secondly, each column of $A(P, N)$ does not have more than two consecutive stretches of 1's (see Proposition 5.1). Here the canonical sequence $S = \mathcal{S}_{G_P} = (1\ 2\ 1\ 3\ 4\ 5\ 2\ 6\ 3\ 4\ 7\ 5\ 6\ 8\ 7\ 8)$. The required substrings of probe neighbors for nonprobe vertices $n_1, n_2, \ldots, n_6$ are $(5\ 2\ 6)$, $(3\ 4\ 5\ 2\ 6\ 3\ 4)$, $(4\ 7\ 5\ 6\ 8\ 7\ 8)$, $(1\ 2\ 1\ 3\ 4\ 5\ 2\ 6\ 3\ 4)$, $\emptyset$, S respectively. Note that G is indeed a PTPIG with an interval representation shown in Table 2 which is constructed by the method described in the sufficiency part of Theorem 4.1.*

5. Further structural properties of PTPIG

In this section we give further structural properties of PTPIG which we require for the recognition algorithm.

Proposition 5.1. *Let $G = (P, N, E)$ be a PTPIG. Then for any canonical ordering of the vertices belonging to P each column of $A(P, N)$ cannot have more than two consecutive stretches of 1's.*

Proof. Let us prove by contradiction. Consider a canonical ordering of vertices belonging to P, say, $\{u_1, u_2, \ldots, u_m\}$. Let w_j be a vertex in N such that in the matrix $A(P, N)$ the column corresponding to w_j has at least three consecutive stretches of 1's. That is, there are five vertices in P, say $u_{i_1}, u_{i_2}, u_{i_3}, u_{i_4}$ and u_{i_5} (with $i_1, i_2, i_3, i_4, i_5 \in \{1, 2, \ldots, m\}$) such that $i_1 < i_2 < i_3 < i_4 < i_5$ and u_{i_1}, u_{i_3} and u_{i_5} are neighbors of w_j while u_{i_2} and u_{i_4} are not neighbors of w_j. Let us prove its impossibility. We prove it case by case.

Let the interval corresponding to the vertex v_{i_k} be $I_{v_{i_k}} = [\ell_k, r_k]$ for $k = 1, 2, 3, 4, 5$ in a PTPIG representation. Then by Theorem 2.1, we have $\ell_1 < \ell_2 < \ell_3 < \ell_4 < \ell_5$ and $r_1 < r_2 < r_3 < r_4 < r_5$. Since G is a PTPIG, either $\ell_i \in I_{w_j}$ or $r_i \in I_{w_j}$ for each $j = 1, 3, 5$.

- Case 1: $(\ell_1, \ell_5 \in I_{w_j})$. In this case, for t such that $i_1 \leq t \leq i_5$, we have $\ell_t \in I_{w_j}$. In particular we have ℓ_2 and ℓ_4 are in I_{w_j}, i.e., u_{i_2} and u_{i_4} are neighbors of w_j which is a contradiction.
- Case 2: $(r_1, r_5 \in I_{w_j})$. In this case, for all t such that $i_1 \leq t \leq i_5$, we have $r_t \in I_{w_j}$. And again here we have a contradiction just like the previous case.
- Case 3: $(\ell_1, r_5 \in I_{w_j}$ but $r_1, \ell_5 \notin I_{w_j})$. Let I_{w_j} be $[\ell_{w_j}, r_{w_j}]$. Thus in this case, $\ell_5 < \ell_{w_j} \leqslant \ell_1$ which is a contradiction.
- Case 4: $(r_1, \ell_5 \in I_{w_j})$. If $\ell_3 \in I_{w_j}$, then $\ell_t \in I_{w_j}$ for all $t \in \{i_3, \ldots, i_5\}$ and this would mean that $\ell_4 \in I_{w_j}$ which is a contradiction. Similarly, if $r_3 \in I_{w_j}$, then $r_t \in I_{w_j}$ for all $t \in \{i_1, \ldots, i_3\}$ and then $r_2 \in I_{w_j}$ which also gives a contradiction.

Note that all the cases are taken care of and thus each column of $A(P, N)$ cannot have more than two consecutive stretches of 1's. □

There are examples in [12] showing that the condition in the above lemma is not sufficient.

Definition 5.2. *Let $G = (V, E)$ be a graph with an independent set N and $P = V \setminus N$ such that G_P, the subgraph induced by P is a proper interval graph. Let $\mathcal{S}_{G_P}$ be a canonical sequence of G_P. Let $w \in N$. If there exists a substring in $\mathcal{S}_{G_P}$ which contains all the neighbors of w and all the vertices in the substring are neighbors of w then we call the substring a* perfect substring of w. *If the canonical sequence $\mathcal{S}_{G_P}$ contains a* perfect substring of w in $\mathcal{S}_{G_P}$ *for all $w \in N$, we call it a* perfect canonical sequence for G.

The following Proposition shows that unless the trivial case no nonprobe vertex has more than one disjoint perfect substrings.

Proposition 5.3. *Let $G = (P, N, E)$ be a PTPIG such that G_P is a connected reduced proper interval graph and $\mathcal{S}_{G_P}$ be a canonical sequence of G_P. Then for any nonprobe vertex $w \in N$, there cannot exist more than one disjoint perfect substring of w in $\mathcal{S}_{G_P}$, unless the substring consists of a single element.*

Proof. Let $u_1, u_2, \ldots, u_p$ be the canonical ordering of the probe vertices of G with the proper interval representation $\{[\ell_i, r_i] \mid i = 1, 2, \ldots, p\}$ that satisfies the condition 4 of Theorem 2.1 and S be the corresponding canonical sequence $\mathcal{S}_{G_P}$. We first note that, since each vertex in S appears twice, there cannot be more than two disjoint perfect substrings of S.

Suppose there is a nonprobe vertex w in G such that there are two disjoint perfect substrings of length greater than 1. We will refer to them as the first substring and the second substring corresponding to the relative location of the substrings in S. In S, each number i appears twice due to l_i and r_i only. Thus if we think of the canonical sequence as an ordering of ℓ_i's and r_i's, then we have that the first substring contains all the ℓ_i's and the second substring contains all the r_i's for all the probe vertices u_i those are neighbors of w, as $\ell_i < r_i$ and both substrings contain all numbers i such that u_i is a neighbor of w.

Moreover due to the increasing order of ℓ_i's and r_i's, both substrings contain numbers $k, k+1, \ldots, k+r$ for some integers k, r with $1 \leqslant k \leqslant m$ and $1 \leqslant r \leqslant m-k$. Then the first substring must comprise of some consecutive collection of ℓ_i and similarly for the second substring, i.e., the first substring is $\ell_k, \ell_{k+1}, \ldots, \ell_{k+r}$ and the second substring is $r_k, r_{k+1}, \ldots, r_{k+r}$ (in $\mathcal{I}_{G_P}$). Therefore the vertices $u_k, \ldots, u_{k+r}$ form a clique.

Suppose u_i is adjacent to u_{k+t} for some $i < k$ and $1 \leq t \leq r$. Then $\ell_i < \ell_k$ and $\ell_{k+r} < r_i$ as ℓ_k to ℓ_{k+r} are consecutive in the first substring (in $\mathcal{I}_{G_P}$). But this implies u_i is adjacent to all $u_k, u_{k+1}, \ldots, u_{k+r}$. Similarly, one can show that if u_j is adjacent to u_{k+t} for some $j > k+r$ and $1 \leq t \leq r$. Then u_j is adjacent to all $u_k, u_{k+1}, \ldots, u_{k+r}$. Thus (closed) neighbors of

$u_k, u_{k+1}, \ldots, u_{k+r}$ are same in G_P which contradicts the assumption that G_P is reduced as $r \geq 1$. □

In fact, we can go one step more in understanding the structure of a PTPIG. If G is a PTPIG, not only there cannot be two disjoint perfect substrings (of length more than 1) for any nonprobe vertex in any canonical sequence but also any two perfect substrings for the same vertex must intersect at least two places, except two trivial cases.

Proposition 5.4. *Let $G = (P, N, E)$ be a PTPIG such that G_P is a connected reduced proper interval graph with a canonical ordering of vertices $\{u_1, u_2, \ldots, u_p\}$ and let $\mathcal{V}_{G_P}$ be the corresponding vertex canonical sequence of G_P. Let $w \in N$ be such that w has at least two neighbors in P and T_1, T_2 be two perfect substrings for w in $\mathcal{V}_{G_P}$ intersecting in exactly one place. Then one of the following holds:*

(1) *$\mathcal{V}_{G_P}$ begins with $u_1u_2u_1$ and only u_1 and u_2 are neighbors of w.*
(2) *$\mathcal{V}_{G_P}$ ends with $u_pu_{p-1}u_p$ and only u_{p-1} and u_p are neighbors of w.*

Proof. Let $[a_i, b_i]$ be the interval corresponding to u_i for $i = 1, 2, \ldots, p$. Let the place where T_1 and T_2 intersect be the first occurrence of the vertex u_k.

Without loss of generality, let the substring T_1 end with the first occurrence of u_k and the substring T_2 start with the first occurrence of u_k. Thus for all $i > k$, the vertex u_i cannot appear before the first occurrence of u_k in the $\mathcal{V}_{G_P}$. Therefore T_1 does not contain any u_i such that $i > k$. Thus w is not a neighbor of any u_i such that $i > k$. Note that, it also means that for any vertex in the neighbor of w (except for u_k) the substring T_1 contains the first occurrence, while the substring T_2 contains the second occurrence. Thus the vertices in the neighborhood of w has to be consecutive vertices in the canonical ordering of G_P. Let the vertices in the neighborhood of w be $u_{k-r}, \ldots, u_{k-1}, u_k$, where $1 \leq r \leq k-1$.

For any vertex u_i such that $i < k - r$, we have u_i is not in T_1 and T_2. So the first occurrence of u_i is before the first occurrence of u_{k-r} and the second occurrence of u_i is either also before the first occurrence of u_{k-r} or after T_2, i.e., after the second occurrence of u_{k-r}. But if the second case happens, then we would violate the fact that G_P is proper. So the only option is that both the first and second occurrence of u_i are before the first occurrence of u_{k-r} and this would violate the condition that the graph G_P is connected. Thus there exists no u_i such that $i < k - r$. This implies $k - r = 1$. Thus we have the neighbors of w precisely $u_1, \ldots, u_k$.

If we look at the interval canonical sequence of G_P, we have T_1 corresponds to $a_1, \ldots, a_k$ and T_2 corresponds to $a_k, b_1, \ldots, b_{k-1}$. But this would mean that all the vertices $u_1, \ldots, u_{k-1}$ have the same (closed) neighborhood in G_P which is not possible as we assumed G_P is reduced, unless the set $\{u_1, \ldots, u_{k-1}\}$ is a single element set. In that case, w has neighbors u_1 and u_2 and the T_1 and T_2 correspond to a_1, a_2 and a_2, b_1 respectively (in $\mathcal{I}_{G_P}$). This is the first option in Proposition 5.4. By a similar argument, if we assume that T_1 and T_2 intersect in the second occurrence of the vertex u_k, we get the other option. □

6. Recognition algorithm

In this section, we present a linear-time recognition algorithm for PTPIG. That is, given a graph $G = (V, E)$, and a partition of the vertex set into N and $P = V \setminus N$ we can check whether the graph $G = (P, N, E)$ is a PTPIG or not in $O(|V| + |E|)$ time. We have $G = (P, N, E)$ is a PTPIG if and only if it is a TPIG, and G_P is a proper interval graph for a TPIG representation of G. Note that it is easy to check in linear time if N is an independent set in the graph. We will use the characterization we obtained in Theorem 4.1 to test if the graph satisfies the other properties.

In order to shorten the length of the paper, the algorithm is briefly outlined here. The full details and pseudo codes are available in [12].

We will employ the recognition algorithm for proper interval graph $H = (V', E')$ given by Booth and Lueker [3] as a black box that runs in $O(|V'| + |E'|)$. The main idea of their algorithm is that H is a proper interval graph if and only if the adjacency matrix of the graph satisfies the consecutive 1's property. So for every vertex v in H, they consider restrictions, on the ordering of the vertices, of the form "all vertices in the neighborhood of v must be consecutive". This is done with the help of the data structure of *PQ*-trees. The PQ-tree helps in storing all the possible orderings that adhere to all these restrictions. It is important to note that all the orderings that satisfy the restrictions are precisely all the canonical orderings of vertices of H.

The key idea behind our recognition algorithm is that if the graph $G = (P, N, E)$ is PTPIG then, from Condition **(A)** in Theorem 4.1, we can obtain a series of restrictions on the ordering of vertices, that also can be "stored" by the use of PQ-tree data structure. These restrictions are on and above the restrictions that we need to ensure the graph G_P is a proper interval graph. Finally, if there exists an ordering of the vertices satisfying all the restrictions, then that ordering will be a canonical ordering that satisfies condition **(A)** in Theorem 4.1. Thus the main challenge is to discover all the additional restrictions on the ordering and how to store them in the PQ-tree.

We first verify that N is an independent set and that the graph G_P is a proper interval graph. In this process, we have stored all the possible canonical ordering of the vertices of the subgraph $G_P = (P, E_1)$ in a PQ-tree (in $O(|P| + |E_1|)$ time). We proceed to find the extra restrictions that must be applied to the orderings.

We present our algorithm in three cases - each case handling a class of graphs that is a generalization of the class of graphs handled in the previous one.

- CASE I: First we consider the case when G_P is a connected reduced proper interval graph.
- CASE II: Next we consider the case when G_P is a connected proper interval graph, but not necessarily reduced.
- CASE III: Finally we consider the general case when the graph G_P is a proper interval graph, but may not be connected or reduced.

For all the cases we will assume that the vertices in P are $v_1, \dots, v_p$ and the vertices in N are $w_1, \dots, w_q$. Let A_j be the adjacency list of the vertex w_j and let d_j be the degree of the vertex w_j. Then the neighbors of w_j are $A_j[1], A_j[2], \dots, A_j[d_j]$ for $j = 1, 2, \dots, q$.

6.1. **Case I: The graph G_P is a connected reduced proper interval graph.**

By Lemma 3.3, there is a unique (up to reversal) canonical ordering of the vertices of G_P. By Theorem 4.1, we know that the graph G is PTPIG if and only if the following condition is satisfied:

Condition (A1): For all $1 \leq j \leq q$, there is a substring in $\mathcal{S}_{G_P}$ where only the neighbors of w_j appear and all the neighbors of w_j appear at least once.

In this case, when the graph G_P is connected reduced proper interval graph, since there is a unique canonical ordering of the vertices, it suffices to check if the corresponding canonical sequence satisfies Condition (A1). The rest of the algorithm in this case is to check if the property is satisfied.

Idea of the algorithm: Since we know the canonical sequence $\mathcal{S}_{G_P}$ (or obtain by using known algorithms described before in $O(|P| + |E_1|)$ time, where E_1 is the set of edges between probe vertices), we form two lookup tables L and R such that for any vertex $v_i \in P$, the $L(v_i)$ and $R(v_i)$ hold the indices of the first and the second appearance of v_i in $\mathcal{S}_{G_P}$ respectively. We can obtain the lookup tables in time $O(|P|)$ steps.

Also by $\mathcal{S}_{G_P}[k_1, k_2]$ (where $1 \leq k_1 \leq k_2 \leq 2p$) we will denote the substring of the canonical sequence $\mathcal{S}_{G_P}$ that start at the k_1^{th} position and ends at the k_2^{th} position in $\mathcal{S}_{G_P}$.

To check Condition (A1), we will go over all the $w_j \in N$. For $j \in \{1, 2, \dots, q\}$, let $L(A_j[1]) = \ell_j$ and $R(A_j[1]) = r_j$. Since all the neighbors of w_j have to be in a substring, there must be a substring of length at least d_j and at most $2d_j$ (as each number appears twice) in $\mathcal{S}_{G_P}[\ell_j - 2d_j, \ell_j + 2d_j]$ or $\mathcal{S}_{G_P}[r_j - 2d_j, r_j + 2d_j]$ which contains only and all the neighbors of w_j. We can find all such substrings by first marking the positions in $\mathcal{S}_{G_P}[\ell_j - 2d_j, \ell_j + 2d_j]$ and $\mathcal{S}_{G_P}[r_j - 2d_j, r_j + 2d_j]$ those are neighbors of w_j and then by doing a double pass, we find all the possible substrings of length greater than or equal to d_j in $\mathcal{S}_{G_P}[\ell_j - 2d_j, \ell_j + 2d_j]$ and $\mathcal{S}_{G_P}[r_j - 2d_j, r_j + 2d_j]$ that contains only neighbors of w_j. Naturally Propositions 5.1, 5.3, and 5.4 play important roles in the construction of the above algorithm.

Going through this way one can correctly decide whether G is a PTPIG with probes P and nonprobes N in time $O(|P| + |N| + |E_2|)$, where E_2 is the set of edges between probes P and nonprobes N when G_P is connected reduced proper interval graph. Since obtaining $\mathcal{S}_{G_P}$ requires $O(|P|+|E_1|)$ time, the total recognition time is $O(|P|+|N|+|E_1|+|E_2|) = O(|V|+|E|)$.

6.2. Case II: The graph G_P is a connected (but not necessarily reduced) proper interval graph.

In this case, since the graph G_P is not reduced, a unique canonical ordering of vertices of G_P may not exist. By Theorem 4.1, all we can say is that among the set of canonical orderings of the vertices of G_P, is there an ordering such that the corresponding canonical sequence satisfies Condition (A) of Theorem 4.1. As mentioned before, we will assume that we have all the possible canonical ordering of the vertices of G_P stored in a PQ-tree. We will impose more constraints on the orderings to satisfy the required condition.

Let $\widetilde{G_P}$ be the reduced graph of G_P. Then $\widetilde{G_P}$ has a unique (up to reversal) canonical ordering of vertices, say, $b_1, \dots, b_t$ (corresponding to the blocks $B_1, \dots, B_t$ of the vertices of G_P) and the canonical orderings of the vertices of G_P are obtained by all possible permutations of the vertices of G within each block. For any $w \in N$ and any block B_k, we say that B_k is a *block-neighbor* of w if there exists at least one vertex in B_k that is a neighbor of w. If all the vertices in B_k are neighbors of w, we call B_k a *full-block-neighbor* of w.

Idea of the algorithm: If G is PTPIG then from condition (A) in Theorem 4.1 we can see that the following condition is a necessary (though not a sufficient) condition:

Condition (B1): For all $1 \leq j \leq q$, there is a substring of $\mathcal{S}_{\widetilde{G_P}}$ where only the block-neighbors of $w_j \in N$ appear. All the block-neighbors of w_j appear at least once, and any block that is not at the beginning or end of the substring must be a full-block-neighbor of w_j (i.e., all vertices of the block are neighbors of w_j).

As condition (B1) is not sufficient for G to be a PTPIG, we need to find a suitable ordering of vertices in each block. We will have a number of cases and for each of the cases, some restrictions will be imposed on the ordering of the vertices within blocks depending upon the neighbors of w_i within each block. We identify all the various kinds of restrictions on $\sigma_1, \dots, \sigma_t$ (orderings of the vertices of blocks $B_1, B_2, \dots, B_t$ respectively) which are necessary to be imposed so that G becomes a PTPIG.

This case is in fact the most technical step. This step crucially uses an algorithm that solves a generalization of the consecutive 1's problem. We call it the *Oriented-consecutive* 1*'s problem.*

Oriented-consecutive 1's problem: As an extension of the consecutive 1's problem, we introduce *Oriented-consecutive* 1*'s problem* that reduces the difficulty of determining if there exist orderings of the vertices that meet all of the constraints to this problem. The PQ-tree can be used to solve the oriented-consecutive $1'$s problem. The *Oriented-consecutive* 1*'s problem* is the following:

Input: A set $\Omega = \{s_1, \ldots, s_m\}$ and n restrictions $(S_1, b_1), \ldots, (S_n, b_n)$ where $S_i \subseteq \Omega$ and $b_i \in \{-1, 0, 1, 2\}$.
Output: All the linear ordering, σ, of Ω such that, in the linear ordering $s_{\sigma(1)}, \ldots, s_{\sigma(m)}$, for $1 \leq i \leq n$ the following are satisfied:

- If $b_i = 0$, then all the elements in S_i are consecutive in the linear ordering.
- If $b_i = -1$, then all the elements in S_i are consecutive in the linear ordering and all the elements of S_i are flushed towards Left, i.e., $s_{\sigma(1)} \in S_i$.
- If $b_i = 1$, then all the elements in S_i are consecutive in the linear ordering and all the elements of S_i are flushed towards Right, i.e., $s_{\sigma(m)} \in S_i$.
- If $b_i = 2$, then all the elements in S_i are consecutive in the linear ordering and all the elements of S_i are either flushed towards Left or flushed towards Right, i.e., either $s_{\sigma(1)} \in S_i$ or $s_{\sigma(m)} \in S_i$.

If $b_i = 2$ implies $b_j = 2$ for all $j \geq i$, then we can design an algorithm that stores all the valid ordering in the PQ-tree T and the amortized running time of the algorithm is linear. The whole algorithm runs in $O(|V| + |E|)$ time.

6.3. **Case III: The graph G_P is a proper interval graph (not necessarily connected or reduced).**

Finally, we consider the graph $G = (V, E)$ with an independent set N (nonprobes) and $P = V \setminus N$ (probes) such that G_P is a proper interval graph, which may not be connected. Let the connected components of G_P be $G_1, \ldots, G_r$ with vertex sets $P_1, P_2, \ldots, P_r$. For G to be a PTPIG, it is essential that the subgraphs of G induced by $P_k \cup N$ is a PTPIG for each $k = 1, 2, \ldots, r$. As we have seen in the last case, we can check if all the subgraphs are PTPIG in time $O(|V| + |E|)$. In fact, for each k, we can store all the possible canonical orderings of vertices in P_k such that the corresponding canonical sequence satisfies condition (A) of Theorem 4.1 so that the graph induced by $P_k \cup N$ is a PTPIG.
Idea of the algorithm: To check if the whole graph G is a PTPIG, we have to find if there exists a canonical ordering of all the vertices in G_P such that for the whole graph, condition (A) of Theorem 4.1 is satisfied. Note that a canonical ordering of the vertices of G_P would place the vertices in each connected component next to each other, and moreover, for each k, the ordering of the vertices of G_k would be a canonical ordering for the graph G_k. In order to check if G is a PTPIG we have to find if there exist an ordering of the connected components and canonical ordering of vertices in each of the components such that the corresponding canonical ordering satisfies condition (A) of Theorem 4.1. In fact, G is a PTPIG if and only if the following condition is satisfied:

Condition (C1): There exists permutation $\pi : \{1, \ldots, r\} \to \{1, \ldots, r\}$ and canonical sequences $\mathcal{S}_{G_1}, \ldots, \mathcal{S}_{G_r}$ of $G_1, \ldots, G_r$ such that the canonical sequence $\mathcal{S}_{G_P}$ of G_P obtained by concatenation of the canonical sequences of $G_{\pi(1)}, \ldots, G_{\pi(r)}$ (that is, $\mathcal{S}_{G_P} = \mathcal{S}_{G_{\pi(1)}} \cdots \mathcal{S}_{G_{\pi(r)}}$) has the property that for all $w \in N$ there exists a *perfect substring* of w in $\mathcal{S}_{G_P}$ (that is, there exists a substring of $\mathcal{S}_{G_P}$ where only the vertices of w appear and all the neighbors of w appears at least once).

Using previous cases, we store all the possible canonical orderings of the vertices in each component so that the graphs induced by $G_k \cup N$ are PTPIG for each k. As usual, we will store the restrictions using the PQ-tree. Next, we will have to add more restrictions on the canonical ordering of the vertices in each of the connected components, which are necessary for the graph G to be a PTPIG. These restrictions will be stored in the same PQ-tree. Finally, we check if an ordering of the components exists such that condition (C1) is satisfied. All this can be done in $O(|V| + |E|)$ time.

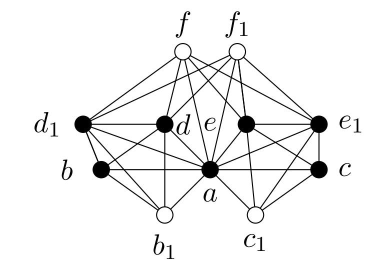
7. PTPIG and its relation with other variants

In this section, we provide the relation between PTPIG and other similar variants. The graphs discussed here are presented in Figure 1. The definition of PIG is very similar to that of TPIG, but the two classes of graphs are not comparable. For example, the graph G_a is PIG, but it is not a TPIG, whereas the graph G_b is a TPIG, but it is not a PIG [14]. But PPIG is a proper subclass of PTPIG as well as of PIG. For example, C_4 with the alternating probe, nonprobe vertices (see the graph G_4) is a PPIG, which is a PIG and a PTPIG with the same interval representation $[1, 4]$, $[5, 8]$ for probe vertices and $[2, 6]$, $[3, 7]$ for the nonprobes.

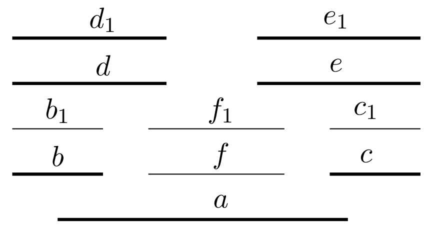
On the other hand, $K_{1,3}$ with a single nonprobe at the center (see the graph G_2) cannot be a PPIG, for otherwise, it would be a proper interval graph (as any probe interval graph with a single nonprobe vertex is an interval graph). But it is a PTPIG by choosing three disjoint intervals for probe vertices and an interval containing all of them corresponding to the nonprobe vertex. As $K_{1,3}$ is an interval graph, G_2 is an example of PIG and PTPIG, but not a PPIG.

Similarly, C_4 with a single nonprobe vertex (see the graph G_3) is a PTPIG with an interval representation $[3, 4]$ for the nonprobe and $\{[1, 3], [2, 5], [4, 6]\}$ for probes, but this is not a PIG (for otherwise it would be an interval graph). Next, we consider the graph G_1. It is a PIG, and TPIG follows from the interval representation described in the figure. But G_1 is not a PTPIG as the subgraph induced by probe vertices is $K_{1,3}$, which is not a proper interval graph.

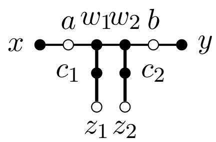
Finally, it is interesting to note that there are examples of TPIG, G for which G_P is a proper interval graph, but G is not a PTPIG. For example, the graph G_b in [14] is a TPIG in which $(G_b)_P$ consists of a path of length 4



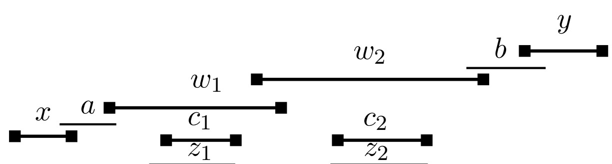
G_a



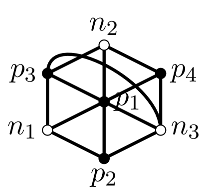
PIG representation of the graph G_a [14]



G_b

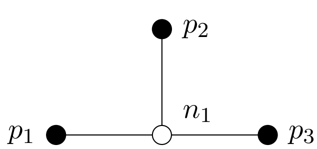


TPIG representation of the graph G_b [14]

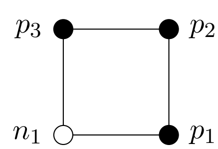


G_1

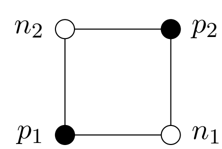
TPIG and PIG representation of the graph G_1



G_2



G_3



G_4

along with 2 isolated vertices, which is a proper interval graph. But G_b has no TPIG representation with a proper interval representation of $(G_b)_P$.

- PPIG ⊂ PTPIG ⊂ TPIG and PPIG ⊂ PIG

Graph	TPIG	PTPIG	PPIG	PIG
G_4	✓	✓	✓	✓
G_2	✓	✓	✗	✓
G_3	✓	✓	✗	✗
G_1	✓	✗	✗	✓
G_a	✗	✗	✗	✓
G_b	✓	✗	✗	✗

FIGURE 1. The relation between graph classes TPIG, PTPIG, PPIG and PIG.

Acknowledgement

We are grateful to the learned referee for valuable suggestions and corrections which improved and enhanced the paper.

References

1. A. Basu, S. Das, S. Ghosh, and M. Sen, *Circular-arc bigraphs and its subclasses*, J. Graph Theory **73** (2013), 361–376.
2. S. Benzer, *On the topology of the genetic fine structure*, Proc. Nat. Acad. Sci. USA **45** (1959), 1607–1620.
3. K. S. Booth and G. S. Leuker, *Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms*, J. Comput. System Sci. **13** (1976), 335–379.
4. D. E. Brown, *Variations on interval graphs*, Ph.D. Thesis, Univ. of Colorado at Denver, USA, 2004.
5. D. G. Corneil, *A simple 3-sweep LBFS algorithm for the recognition of unit interval graphs*, Discrete Appl. Math. **138** (2004), 371–379.
6. X. Deng, P. Hell, and J. Huang, *Linear time representation algorithms for proper circular arc graphs and proper interval graphs*, SIAM J. Comput. **25** (1996), 390–403.
7. S. Ghosh, M. Podder, and M. Sen, *Adjacency matrices of probe interval graphs*, Discrete Appl. Math. **158** (2010), 2004–2013.
8. M. C. Golumbic, *Algorithmic graph theory and perfect graphs*, Annals of Disc. Math., **57**, Elsevier Sci., USA, 2004.
9. P. Hell, R. Shamir, and R. Sharan, *A fully dynamic algorithm for recognizing and representing proper interval graphs*, Siam J. Comput. **31** (2001), 289–305.
10. F. McMorris, C. Wang, and P. Zhang, *On probe interval graphs*, Discrete. Appl. Math. **88** (1998), 315–324.
11. Y. Nussbaum, *Recognition of probe proper interval graphs*, Discrete Appl. Math. **167** (2014), 228–238.
12. S. Paul, S. Ghosh, S. Chakraborty, and M. Sen, *Characterization and recognition of proper tagged probe interval graphs*, arxiv:1607.02922[math.CO], 2016.
13. T. Saitoh, K. Yamanaka, M. Kiyomi, and R. Uehara, *Random generation and enumeration of proper interval graphs*, WALCOM: Algorithm and Computation: Third International Workshop, WALCOM 2009, Kolkata. Ed. S. Das and R. Uehara, LNCS **5431** (2009), 177–189.

14. L. Sheng, C. Wang, and P. Zhang, *Tagged probe interval graphs*, J. Combinatorial optimization **5** (2001), 133–142.
15. ______, *On the Perfectness of Tagged Probe Interval Graphs*, Discrete Mathematical Problems with Medical Applications, American Mathematical Society, Providence, RI, 159–163, 2000.
16. ______, *Tagged probe interval graphs*, DIMACS Technical Report 98-12, 1998.
17. P. Zhang, *Probe interval graphs and their application to physical mapping of DNA*, Manuscript, 1994.

Department of Mathematics, Jadavpur University, Kolkata - 700 032, India.
E-mail address: sanchitajumath@gmail.com

Department of Mathematics, Jadavpur University, Kolkata - 700 032, India.
E-mail address: ghoshshamik@yahoo.com (Corresponding author)

Indian Statistical Institute, Kolkata - 700 108, India.
E-mail address: sourav@isical.ac.in

Department of Mathematics, North Bengal University, District - Darjeeling, West Bengal, Pin - 734 430, India.
E-mail address: senmalay10@gmail.com