Characterization and recognition of proper tagged probe interval graphs

Sourav Chakraborty[1], Shamik Ghosh[2], Sanchita Paul[3], and Malay Sen[4]

[1] Indian Statistical Institute, Kolkata - 700108, India
sourav@isical.ac.in
[2] Depratment of Mathematics, Jadavpur University, Kolkata-700032, India
ghoshshamik@yahoo.com
[3] Depratment of Mathematics, Jadavpur University, Kolkata-700032, India
sanchitajumath@gmail.com
[4] Department of Mathematics, North Bengal University, West Bengal - 734430, India. senmalay10@gmail.com

Abstract. Interval graphs were used in the study of genomics by the famous 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. A tagged probe interval graph (briefly, TPIG) is motivated by similar applications to genomics, where the set of vertices is partitioned into two sets, namely, probes and nonprobes and there is an interval on the real line corresponding to each vertex. The graph has an edge between two probe vertices if their corresponding intervals intersect, has an edge between a probe vertex and a nonprobe vertex if the interval corresponding to a nonprobe vertex contains at least one end point of the interval corresponding to a probe vertex and the set of nonprobe vertices is an independent set. This class of graphs have been defined nearly two decades ago, but till today there is no known recognition algorithm for it.

In this paper, we consider a natural subclass of TPIG, namely, the class of *proper tagged probe interval graphs* (in short PTPIG). We present characterization and a linear time recognition algorithm for PTPIG. To obtain this characterization theorem we introduce a new concept called *canonical sequence* for proper interval graphs, which, we belief, has an independent interest in the study of proper interval graphs. Also to obtain the recognition algorithm for PTPIG, we emphasize on a variation of consecutive 1's problem, namely, *oriented-consecutive 1's problem* and some variations of PQ-tree algorithm.

Keywords: Interval graph, proper interval graph, probe interval graph, probe proper interval graph, tagged probe interval graph, consecutive 1's property, PQ-tree algorithm.

1 Introduction

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. Such a mapping of vertices into an interval on the real line is called an *interval representation of* G. The study of interval graphs was motivated by the study of the famous molecular biologist Benzer [1] in 1959. Since then interval graphs has been widely used in molecular biology and genetics, particularly for DNA sequencing. Different variations of interval graphs have been used to model different scenarios arising in the area of DNA sequencing. Literature on the applications of different variations of interval graphs can be found in [4, 11, 17].

In an attempt to aid a problem called *cosmid contig mapping*, a particular component of the physical mapping of DNA, in 1998 Sheng, Wang and Zhang [26] defined a new class of graphs called *tagged probe interval graphs* (briefly, TPIG) which is a generalization of interval graphs. Since then one of the main open problems in this area has been "Given a graph G, recognizing whether G is a *tagged probe interval graphs*."

Definition 11. A graph $G = (V, E)$ is a *tagged probe interval graph* 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 $I_x = [\ell_x, r_x]$) such that all the following conditions hold:

1. N is an independent set in G, i.e., there is no edge between nonprobe vertices.
2. If $x, y \in P$, then there is an edge between x and y if and only if $I_x \cap I_y \neq \emptyset$, or in other words the mapping is an interval representation of the subgraph of G induced by P.
3. If $x \in P$ and $y \in N$, then there is an edge between x and y if and only if the interval corresponding to the nonprobe vertex contains at least one end point of the interval corresponding to the probe vertex, i.e., either $\ell_x \in I_y$ or $r_x \in I_y$.

We call the collection $\{I_x \mid x \in V\}$ 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 $G = (P, N, E)$.

Problem 1. Given a graph $G = (P, N, E)$, give a linear time algorithm for checking if G is a *tagged probe interval graph.*

Tagged probe inteval graphs have been defined nearly two decades ago and its importance in the context of molecular biology has been emphasized over and over again [25–28]. Yet untill this paper there was no known algorithm for tagged probe interval graphs or any natural subclass of tagged probe interval graphs, excepting probe proper interval graphs.

A natural and well studied subclass of interval graphs are the *proper interval graphs*. A *proper interval graph* G is an interval graph in which there is an interval representation of G such that no interval contains another properly. Such an interval representation is called a *proper interval representation of* G. Proper interval graphs is an extremely rich class of graphs and we have a number of different characterizations of proper interval graphs.

In this paper we study a natural special case of tagged probe interval graphs which we call *proper tagged probe interval graph* (in short, *PTPIG*). The only extra condition that a PTPIG should satisfy over TPIG is that the mapping of the vertices into intervals on the real line that gives a TPIG representation of G should be a **proper interval representation** of the subgraph of G induced by P.

In this paper, we present a linear time (linear in $(|V|+|E|)$ recognition algorithm for PTPIG. The backbone of our recognition algorithm is the characterization of proper tagged probe interval graphs that we obtain in Theorem 33 in Section 3. To obtain this characterization theorem we introduce (in Section 2) a new concept called **"canonical sequence"** for proper interval graphs, which we believe would be of independent interest in the study of proper interval graphs. The concept of canonical sequence for proper interval graphs can be used to solve other problems related to proper interval graphs, for example testing isomorphism of proper interval graphs.

1.1 Organization of the Paper

We present the characterization pf Proper Tagged Probe Interval Graphs (PTPIG) in Section 3. The main combinatorial object we need for the characterization is the Canonical Sequence of Proper Interval Graphs that is presented in Section 2

We present our recognition algorithm in Section 4. In the road to obtain the linear time recognition algorithm for PTPIG, we face a number of algorithmic challenges that led us to solve a number of sub-problems that can be of independent interest. One particular problem worth mentioning is a generalization of the well known consecutive 1's problem, which we call the *oriented-consecutive 1's problem*. It is a very important sub-routine in our recognition algorithm.

Due is lack of space we are unable to provide the technical proofs of various lemmas and theorems in this extended abstract. We do give a brief idea of the proofs wherever possible. The complete proofs will be presented in the journal-version of the paper. A first draft of the proofs can also be found in the ArXiv version of our paper [20].

1.2 Notations

Suppose a graph G is a PTPIG (or TPIG), then we will assume that the vertex set is partitioned into two sets P (for probe vertices) and N (for nonprobe vertices). To indicate that the partition is known to us, we will sometimes denote G by $G = (P, N, E)$, where E is the edge set. We will denote by G_P the subgraph of G that is induced by the vertex set P. We will assume that there are p probe vertices $\{u_1, \ldots, u_p\}$ and q nonprobe vertices $\{w_1, \ldots, w_q\}$. To be consistent in our notation we will use i or i' or $i_1, i_2, \ldots$ as indices for probe vertices and use j or j' or $j_1, j_2, \ldots$ as indices for nonprobe vertices.

Let $G = (V, E)$ be a graph and $v \in V$. Then by the *closed neighborhood* of v in G we mean the set $N[v] = \{u \in V \mid u \text{ is adjacent to } v\} \cup \{v\}$. A graph G is called *reduced* if no two vertices have the same closed neighbourhood. 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 $B(v)$. So, the collection of blocks is a partition of V. The reduced graph of G (denoted $\widetilde{G} = (\widetilde{V}, \widetilde{E})$) is the graph obtained by merging all the vertices that are in the same equivalence class.

If M is a $(0,1)$-matrix, then we say M satisfies the *consecutive* 1*'s property* if in each column 1's appear consecutively. [15, 18] By $A(G)$ we denote the *augmented adjacency matrix* of the graph G, in which the diagonal entries are 1 and non-diagonal entries are same as the adjacency matrix of G.

1.3 Background Materials

PQ-trees: In the past few decades many variations of interval graphs has been studied mostly in context of modeling different scenario arising from molecular biology and DNA sequencing. Understanding the structure and properties of these classes of graphs and designing efficient recognition algorithms are the central problems in this field. Many times this studies have led to nice combinatorial problems and development of important data structures.

For example, the original linear time recognition algorithm for interval graphs by Booth and Lueker [3] in 1976 is based on their complex PQ tree data structure (also see [2]). Habib et al. [12] in 2000 showed how to solve the problem more simply using lexicographic breadth-first search, based on the fact that a graph is an interval graph if and only if it is chordal and its complement is a comparability graph. A similar approach using a 6-sweep LexBFS algorithm is described in Corneil, Olariu and Stewart [7] in 2010.

In this paper we will be using the data structure of PQ-trees quite extensively. PQ-trees are not only used to check whether a given matrix satisfy the consecutive 1's property, they also store all the possible permutations such that if one permuted the rows using the permutation, the matrix would satisfy the consecutive 1's property. We generalize the problem of checking consecutive 1's property to Oriented-consecutive 1's problem and used the PQ-tree representation to solve this problem also.

Proper Interval Graphs: For proper interval graphs we have a number of equivalent characterizations. Recall that a proper interval graph G is an interval graph in which there is an interval representation of G such that no interval contains another properly and such an interval representation is called a *proper interval representation* of G. It is important to note that a proper interval graph G may have an interval representation which is not proper. Linear-time recognition algorithms for proper interval graphs are obtained in [8, 9, 13, 14]. A *unit interval graph* is an interval graph in which there is an interval representation of G such that all intervals have the same length. Interestingly, these two concepts are equivalent. Another equivalence is that an interval graph is a proper interval graph if and only if it does not conatin $K_{1,3}$ as an induced subgraph. Apart from these, there are several characterizations of proper interval graphs (see Table 1). Among them we repeatedly use the following equivalent conditions in the rest of the paper:

Theorem 12. *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).*

Index	Properties	References
1	G is a Proper Interval Graph.	[10, 21–24]
2	G is a Unit Interval Graph.	
3	G is *claw-free*, i.e., G does not contain $K_{1,3}$ as an induced subgraph.	
4	For all $v \in V$, elements of $N[v] = \{u \in V \mid uv \in E\} \cup \{v\}$ are consecutive for some ordering of V (closed neighborhood condition).	[5–7, 10]
5	There is an ordering $v_1, v_2, \cdots, v_n$ of V such that G has a proper interval graph representation $\{I_{v_i} = [a_i, b_i] \| i = 1, 2, \cdots, n\}$ where $a_1 < a_2 < \cdots < a_n$ and $b_1 < b_2 < \cdots < b_n$.	
6	There is an ordering of V such that the augmented adjacency matrix $A(G)$ of G satisfies the consecutive 1's property.	
7	*A* straight enumeration *of* G *is a linear ordering of blocks (vertices having same closed neighborhood) in* G, *such that for every block, the block and its neighboring blocks are consecutive in the ordering.* G has a straight enumeration (which is unique up to reversal, if G is connected).	[8, 13, 14, 19]
8	*The* reduced graph $\widetilde{G}$ *is obtained from* G *by merging vertices having same closed neighborhood.* $G(n, r)$ *is a graph with* n *vertices* $x_1, x_2, \ldots, x_n$ *such that* x_i *is adjacent to* x_j *if and only if* $0 < \|i - j\| \leqslant r$, *where* r *is a positive integer.* $\widetilde{G}$ is an induced subgraph of $G(n, r)$ for some positive integers n, r with $n > r$.	[16]

Table 1. Characterizations of proper interval graphs: equivalent conditions on an interval graph $G = (V, E)$.

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

Remark 13. 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.

2 Canonical Sequence of Proper Interval Graphs

Let G be a proper interval graph. Then there is an ordering of V that satisfies conditions (2), (3) and (4) of Theorem 12. Henceforth we call such an ordering, a *natural* or *canonical* ordering of V. But this canonical ordering is not unique. A proper interval graph may have more than one canonical orderings of its vertices. Interestingly, it follows from Corollary 2.5 of [8] (also see [19]) that the canonical ordering is unique up to reversal for a connected reduced proper interval graph.

Definition 21. Let $G = (V, E)$ be a proper interval graph. 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 $b_1 < b_2 < \cdots < b_n$. Now we combine all a_i and b_i $(i = 1, 2, \ldots, n)$ in an increasing sequence which we call the *interval canonical sequence* with respect to the canonical ordering of vertices of G and is denoted by $\mathcal{I}_G$.
Now if we replace a_i or b_i by i for all $i = 1, 2, \ldots, n$ in $\mathcal{I}_G$, then we obtain a sequence of integers belonging to $\{1, 2, \ldots, n\}$ each occurring twice. We call such a sequence a *canonical sequence* of G with respect to the canonical ordering of vertices of G and is denoted by $\mathcal{S}_G$. Moreover, if we replace i by v_i for all $i = 1, 2, \ldots, n$ in $\mathcal{S}_G$, then the resulting sequence is called a *vertex canonical sequence* of G (corresponding to the canonical sequence $\mathcal{S}_G$) and is denoted by $\mathcal{V}_G$.
Note that $\mathcal{S}_G$ and its corresponding $\mathcal{V}_G$ and $\mathcal{I}_G$ can all be obtained uniquely from each other. By abuse of notations, sometimes we will use the term canonical sequence to mean any of these.

2.1 Structure of the Canonical Sequence for Proper Interval Graphs

If a graph G is a connected reduced proper interval graph then the following lemma states that the canonical sequence for G is unique upto reversal.

Lemma 22. *Let $G = (V, E)$ be a proper interval graph and $V = \{v_1, v_2, \ldots, v_n\}$ be a canonical ordering of vertices of G. Then the canonical sequence $\mathcal{S}_G$ is independent of proper interval representations that satisfies the given canonical ordering. Moreover $\mathcal{S}_G$ is unique up to reversal for connected reduced proper interval graphs.*

Now there is an alternative way to get the canonical sequence directly from the augmented adjacency matrix. 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*) and U contains precisely all the zeros right to the principal diagonal of $A(G)$. This is possible due to the consecutive 1's property of $A(G)$. Now 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 *stair sequence* of $A(G)$ and note that it is same as the canonical sequence of G with respect to the given canonical ordering of vertices of G.

Proposition 23. *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 canonical sequence $\mathcal{S}_G$ of G is the same as the stair sequence of $A(G)$.*

Corollary 24. *If G is a connected proper interval graph then $\mathcal{S}_G$ is unique up to reversal.*

Remark 25. Let G be a connected proper interval graph which is not reduced and $\widetilde{G}$ be the reduced graph of G. Then the graph $\widetilde{G}$ has a unique (upto reversal) canonical ordering of vertices, say, $b_1, \ldots, b_t$ (corresponding to the blocks $B_1, \ldots, B_t$) as it is connected and reduced. Now the canonical orderings of the vertices of G are obtained from this ordering (and its reversal) by all possible permutation of the vertices of G within each block. In all such cases $\mathcal{S}_G$ will remain same up to reversal.

3 Structure of PTPIG

Let us recall the definition of proper tagged probe interval graph.

Definition 31. 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 the graph G_P. We call such an interval representation a *PTPIG representation* of G.

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 (see Figure 1) in [26] is a TPIG in which $(G_b)_P$ consists of a path of length 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 $(G_b)_P$.

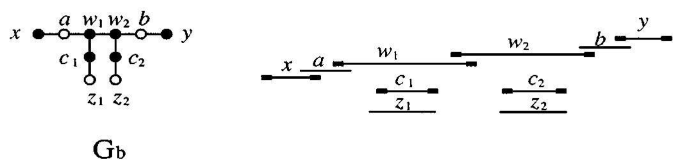

Fig. 1. The graph G_b and its TPIG representation [26]

Now 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. Now 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. In the following lemma we obtain a necessary condition for a PTPIG.

Lemma 32. *Let $G = (P, N, E)$ be a PTPIG. Then for any canonical ordering of the vertices belonging to P each column of $A(P,N)$ can not have more than two consecutive stretches of* 1*'s.*

Unfortunately the condition in the above lemma is not sufficient. For convenience, we say an interval $I_p = [a, b]$ *contains strongly* an interval $I_n = [c, d]$ if $a < c \leqslant d < b$, where $p \in P$ and $n \in N$.[5] The following is a characterization

[5] In [28], Sheng et al. used the term "contains properly" in this case. Here we consider a different term in order to avoid confusion with the definition of proper interval graph. Note that if $a \leqslant c \leqslant d < b$ or $a < c \leqslant d \leqslant b$, then also I_p contains I_n properly, but not strongly.

theorem for a PTPIG and is our main Theorem. For convenience, henceforth, a continuous stretch (subsequence) in a canonical sequence will be called a *substring.*

Theorem 33. *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.*

Such a substring will be called a *perfect substring* (cf. Definition 35).

Idea of the proof: As G_P is a proper interval graph, there exist a canonical ordering of vertices of P satisfying conditions of Theorem 12. Hence one can get a canonical sequence S_{G_P} from this ordering which is basically a combined increasing sequence of the endpoints of the probe intervals. From the definition of PTPIG we know that adjacency between a probe and a nonprobe vertex happens when interval corresponding to the nonprobe vertex intersects interval corresponding to the probe vertex to one of its ends. Hence the end points of the neighbours of a nonprobe vertex $w \in N$ must occur consecutively in S_{G_P}. As all the vertices which are neighbours of $w \in N$ occur consecutively as a substring in S_{G_P}, one can able to construct interval corresponding to w by taking first and last positions of the substring as its endpoint. Note that each probe vertex occurs twice in S_{G_P}. Hence one can assign the intervals corresponding to the probe vertices by taking their first and second occurrence positions in S_{G_P} as their end points.

Remark 34. 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 at the beginning of Section 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.

We conclude this section with some nice structural perception of the substrings described in condition (A) of Theorem 33.

Definition 35. Let $G = (V, E)$ be a graph with an independent set N and $P = V \smallsetminus N$ 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 in G*. 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*.

Proposition 36. *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 substrings of w in $\mathcal{S}_{G_P}$, unless the substring consists of a single element.*

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 atleast two places, except two trivial cases.

Lemma 37. *Let $G = (P, N, E)$ be a PTPIG such that G_P is a connected reduced PIG 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 and 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 $u_1 u_2 u_1$ and only u_1 and u_2 are neighbors of w.*
2. *$\mathcal{V}_{G_P}$ ends with $u_p u_{p-1} u_p$ and only u_{p-1} and u_p are neighbors of w.*

4 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 $P = V \smallsetminus N$ we can check, in time $O(|V| + |E|)$, if the graph $G(P, N, E)$ is a PTPIG. Now $G = (P, N, E)$ is a PTPIG if and only if it a TPIG, i.e., it satisfies the three conditions in Definition 11 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 the graph G satisfies the first condition, namely, if N is an independent set in the graph. Now for testing if the graph satisfies the other two properties we will use the characterization we obtained in Theorem 33.

We will use the recognition algorithm for proper interval graph $H = (V', E')$ given by Booth and Lueker [3] as a blackbox 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. In other words, H is a proper interval graph if and only if there is an ordering of the vertices of H such that for any vertices v in H, the neighbors of v must be consecutive in that ordering. 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 using the data structure of PQ-trees. The PQ-tree helps to store all the possible orderings that respect all these kind of restrictions. It is important to note that all the orderings that satisfy all the restrictions are precisely all the canonical orderings of vertices of H.

The main idea of our recognition algorithm is that if the graph $G = (P, N, E)$ is PTPIG then, from Condition **(A)** in Theorem 33, we can obtain a series of restrictions on the ordering of vertices that also can be "stored" using the 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. If finally there exists any ordering of the vertices that satisfy all the restrictions, then that ordering will be a canonical ordering that satisfies the condition **(A)** in Theorem 33. So the main challenge is to identify all the extra restrictions on the ordering and how to store them in the PQ-tree.

Once we have verified that the N is an independent set and the graph G_P is a proper interval graph and 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 is necessary to be applied on the orderings. We present our algorithm in 3 steps - each step handling a class of graphs that is a generalization of the class of graphs handled in the previous one.

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

4.1 Step I: The graph G_P is a connected reduced proper interval graph

By Lemma 22, there is a unique (up to reversal) canonical ordering of the vertices of G_P. By Theorem 33, 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, all we have to do is to check if the corresponding canonical sequence satisfies the Condition (A1). So 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 can have two look up tables L and R such that for any vertex $v_i \in P$, the $L(v_i)$ and $R(v_i)$ has the index of the first and the second appearance of v_i in $\mathcal{S}_{G_P}$ respectively. We can obtain the look up 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 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 the Condition (A1), we will go over all the $w_j \in N$. For $j \in \{1, 2, \ldots, q\}$, let $L(A_j[1]) = \ell_j$ and $R(A_j[1]) = r_j$. Now 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 identify all such possible 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.

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. As obtaining $\mathcal{S}_{G_P}$ requires $O(|P| + |E_1|)$ time, the total recognition time is $O(|P| + |N| + |E_1| + |E_2|) = O(|V| + |E|)$.

4.2 Step II: The graph G_P is a connected (not necessarily reduced) proper interval graph

In this case, that is when the graph G_P is not reduced, we cannot say that there exists a unique canonical ordering of the vertices of G_P. By Theorem 33, 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 33. 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. Now we will impose more constraints on the orderings so that the required condition is satisfied.

Let $\widetilde{G_P}$ be the reduced graph of G_P. By Remark 25, $\widetilde{G_P}$ has a unique (upto reversal) canonical ordering of vertices, say, $b_1, \ldots, b_t$ (corresponding to the blocks $B_1, \ldots, B_t$ 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 G to be a PTPIG we will have to find a suitable canonical ordering of the vertices or in other words, by Remark 25, we need to find suitable ordering of vertices in each block such that the Condition (A) from Theorem 33 is satisfied. Using the structure of the canonical sequence of $\widetilde{G_P}$ we identity different cases and each case we can identify the various restrictions on the ordering of the vertices in each block that are necessary and sufficient. A generalization of the consecutive 1's problem called the *Oriented-consecutive 1's problem* has been introduced keeping in mind all the restrictions to this problem. We can solve the oriented-consecutive 1's problem using the PQ-tree.

4.3 Step 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 $G_1, \ldots, G_r$. Now, 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, $\mathcal{S}_{G_P} = \mathcal{S}_{G_{\pi(1)}} \ldots \mathcal{S}_{G_{\pi(r)}}$) 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).

Firstly using previous steps we store all the possible canonical orderings of the vertices in each component so that the graphs induced by $G_k \cup N$ is a PTPIG, for each k. As usual we will store the restrictions using the PQ-tree. Next we will have to add some 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 restriction will be be stored in the same PQ-tree. At last we check if there exists an ordering of the components such that the Condition (C1) is satisfied.

5 Conclusion

The study of interval graphs was spearheaded by Benzer [1] in his studies in the field of molecular biology. In [29], Zhang introduced a generalization of interval graphs called probe interval graphs (PIG) in an attempt to aid a problem called cosmid contig mapping. In order to obtain a better model another generalization of interval graphs were introduced that capture the property of overlap information, namely tagged probe interval graphs (TPIG) by Sheng, Wang and Zhang in [26]. Still there is no recognition algorithm for TPIG, in general.

In this paper we characterize and obtain linear time recognition algorithm for a special class of TPIG, namely proper tagged probe interval graphs (PTPIG). The problem of obtaining a recognition algorithm for TPIG, in general is challenging and open till date. It is well known that an interval graph is a proper interval graph if and only if it does not contain $K_{1,3}$ as an induced subgraph of it. Similar forbidden subgraph characterization for PTPIG is another interesting problem.

References

1. Benzer, .S: On the topology of the genetic fine structure, Proc. Nat. Acad. Sci. USA **45**, 1607–1620 (1959).
2. Booth, K. S. and Leuker, G. S.: Linear algorithm to recognize interval grapha and test for the consecutive ones property, Proc. 7th ACM Symp. Theory of computing, 255–265 (1975).
3. Booth, K. S. and Leuker, G. S.: Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms, J. Comput. System Sci., **13**, 335–379 (1976).
4. Brown, D. E.: Variations on interval graphs, Ph.D. Thesis, Univ. of Colorado at Denver, USA (2004).

5. Corneil, D.G.: A simple 3-sweep LBFS algorithm for the recognition of unit interval graphs, Discrete Appl. Math. **138**, 371 - 379 (2004).
6. Corneil, D.G., Olariu, S., Stewart, L.: The LBFS structure and recognition of interval graphs, in preparation; extended abstract appeared as The ultimate interval graph recognition algorithm?, in: Proceedings of SODA 98, Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, USA, , pp. 175–180 (1998).
7. Corneil, D.G., Olariu, S., Stewart, L.: The LBFS structure and recognition of interval graphs, SIAM J. Discrete Math., 23 1905–1953 (2010), .
8. Deng, X., Hell, P. and Huang, J.: Linear time representation algorithms for proper circular arc graphs and proper interval graphs, SIAM J. Comput. **25** 390–403 (1996).
9. de Figueiredo, C.M.H., Meidanis, J. and de Mello, C.P.: A linear-time algorithm for proper interval graph recognition, Inform. Process. Lett. **56** 179–184 (1995).
10. Golumbic, M. C.: Algorithmic graph theory and perfect graphs, Annals of Disc. Math., **57**, Elsevier Sci., USA (2004).
11. Golumbic, M. C. and Trenk, A.: Tolerence Graphs, Cambridge studies in advanced mathematics, Cambridge University Press (2004).
12. Habib, M., McConnel, R., Paul, C. and Viennot, L.: Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition, and consecutive ones testing, Theor. Comput. Sci. **234** 59—84 (2000).
13. Hell, Pavol. and Huang, Jing.: Certifying LexBFS Recognition Algorithms for Proper Interval Graphs and Proper Interval Bigraphs, SIAM J. Discret. Math. **18** 554–570 (2005).
14. Hell, P., Shamir R., and Sharan, R.: A fully dynamic algorithm for recognizing and representing proper interval graphs, SIAM J. Comput. **31** 289–305 (2001).
15. Hsu, Wen-Lian.: A Simple Test for the Consecutive Ones Property, J. Algorithms, **43** 1–16 (2002).
16. Malik, D. S., Sen, M. K. and Ghosh, S.: Introduction to Graph Theory, Cengage Learning, New York, (2014).
17. McKee, Terry A. and McMorris, F. R.: Topics in intersection graph theory, Siam Monographs on Discrete Mathematics and Applications, Philadelphia (1999).
18. Meidanis, J., Porto, O., Telles, G. P.: On the consecutive ones property, Disc. App. Math. **88**, 325-354 (1998).
19. Nussbaum, Y.: Recognition of probe proper interval graphs, Discrete Applied Mathematics, **167** 228–238 (2014).
20. Paul, S., Ghosh, S., Chakraborty, S. and Sen M.: Characterization and recognition of proper tagged probe interval graphs, ArXiv: https://arxiv.org/abs/1607.02922.
21. Roberts, F.S.: Representations of indifference relations, Ph. D. Thesis, Stanford Univeristy (1968).
22. Roberts, F.S.: Indifference graphs, in: F. Harary (Ed.), Proof Techniques in Graph Theory, Academic Press, New York, pp. 139–146 (1969).
23. Roberts, F.S.: On the compatibility between a graph and a simple order, J. Combin. Theory Ser. B **11** 28–38 (1971).
24. Roberts, F.S.: Some applications of graph theory, Chapter in book Combinatorial and Computational Mathematics, Published by World Scientific Publishing Co. Pte. Ltd.
25. Sheng, L.: Cycle free probe interval graphs, Congressus Numerantium, **140** 33–42 (1999).
26. Sheng, L., Wang, C. and Zhang, P.: Tagged probe interval graphs, J. Combinatorial optimization, **5** 133–142 (2001).

27. Sheng, L., Wang, C. and Zhang, P.: On the Perfectness of Tagged Probe Interval Graphs (with Chi Wang, Peisen Zhang, Discrete Mathematical Problems with Medical Applications, American Mathematical Society, Providence, RI, 159–163 (2000).
28. Sheng, L., Wang, C., and Zhang, P.: Tagged probe interval graphs, DIMACS Technical Report 98-12 (1998).
29. Zhang, P.: Probe interval graphs and their application to physical mapping of DNA, Manuscript (1994).