Linear-Time Recognition of Probe Interval Graphs

Ross M. McConnell[1] and Yahav Nussbaum[2]

[1] Computer Science Department, Colorado State University,
Fort Collins, CO 80528, USA
rmm@cs.colostate.edu
[2] The Blavatnik School of Computer Science, Tel Aviv University,
69978 Tel Aviv, Israel
yahav.nussbaum@cs.tau.ac.il

Abstract. The interval graph for a set of intervals on a line consists of one vertex for each interval, and an edge for each intersecting pair of intervals. A probe interval graph is a variant that is motivated by an application to genomics, where the intervals are partitioned into two sets: probes and non-probes. The graph has an edge between two vertices if they intersect and at least one of them is a probe. We give a linear-time algorithm for determining whether a given graph and partition of vertices into probes and non-probes is a probe interval graph. If it is, we give a layout of intervals that proves that it is. In contrast to previous algorithms for the problem, our algorithm can determine whether the layout is uniquely constrained. As part of the algorithm we solve the consecutive-ones probe matrix problem.

1 Introduction

An *interval graph* is the intersection graph of a set of intervals on a line. The set of intervals constitutes an *interval model* of the graph. Interval graphs play an important role in many problems, see [5,7,9]. The problem of recognizing whether a graph is an interval graph played a key role in the 1950's in proving the linear topology of DNA [1]; the intervals were fragments of genetic material, and it was shown empirically that their intersections form an interval graph.

This gave rise to interest in algorithms for determining whether a graph is an interval graph [6]. Booth and Lueker gave the first linear-time algorithm for recognizing interval graphs and constructing interval models for the graphs in the 1970's [2]. A *consecutive-ones ordering* of columns of a 0-1 matrix is one such that, for every row, the 1's in the row are consecutive. Booth and Lueker's approach was to reduce the problem to that of finding a consecutive-ones ordering of a 0-1 matrix, and to give a linear time bound for finding such an ordering.

A related application for interval graphs is *physical mapping*, which can be used for DNA sequencing. In this process, biologists create *clones*, which are copies of fragments of DNA. The problem is reconstruction of the arrangement of the clones in the genome. For some clones, called *probes*, the intersection data between them and other clones can be collected. If all clones are probes, then

A. Fiat and P. Sanders (Eds.): ESA 2009, LNCS 5757, pp. 349–360, 2009.

we can construct an interval graph from the clones, and an interval model for this graph gives the original sequence.

A *probe interval graphs* [14,17] (also called interval probe graph) is a graph in which the vertex set is partitioned into *probes* and *non-probes*. It is a generalization of intersection graph of an interval model, such that the graph has an edge between two vertices if their intervals intersect *and at least one of them is a probe*. Information about which pairs of non-probe intervals intersect is missing.

There has been recent work on topological and combinatorial properties of these graphs; see [9] for a survey. The problem of recognizing whether a graph is a probe interval graph, and finding a corresponding arrangement of intervals if it is, was first shown to be polynomial by Johnson and Spinrad [10], who gave an $O(V^2)$ algorithm. Using a different approach, McConnell and Spinrad gave an $O(V + E \log V)$ algorithm [13]. The latter algorithm was a critical step in the first linear-time algorithm for recognizing *circular-arc graphs* [12]. Motivated by the biological application, where the partition into probes and non-probes is known in advance, both algorithms get as an input a graph whose vertex set is partitioned into probes and non-probes. Chang et al. [4] consider the problem of recognizing this graph class when this partition is not given.

In this paper, we give the first linear-time algorithm for recognizing whether a graph is a probe interval graph when the partition into probes and non-probes is given, and for finding a corresponding set of intervals. In view of the complexity of the previous work, it is surprising that we are able to reduce the problem to that of finding consecutive-ones orderings of two easily-constructed consecutive-ones matrices, which can then be solved by Booth and Lueker's algorithm.

In the physical mapping problem, the arrangement of clones on the genome is certain to be reconstructed accurately only if there is a unique linear arrangement that is consistent with the probe interval graph. Previous algorithms for finding probe interval arrangements have the defect that they cannot determine whether the graph uniquely constrains the arrangement, and therefore cannot be said to solve the physical mapping problem. Uehara [16] has addressed the issue and gave a polynomial-time algorithm that determines whether a given probe interval graph has a unique model. Our algorithm solves this problem as a by-product of the recognition problem, and it is the first linear-time algorithm that solves it.

A 0-1 matrix is a *consecutive-ones matrix* if it has a consecutive-ones ordering. The consecutive-ones sandwich problem is an extension of this problem where the matrix has 0, 1 or $*$. A $*$ is a "don't care"; it can stand for either a 0 or a 1. This problem is NP-Complete [8]. If we require that the $*$'s form a submatrix then we get the consecutive-ones probe matrix problem (see also [3]). We solve this problem in linear time, for any 0, 1, $*$ probe matrix.

2 Preliminaries

Except for some additional definitions, we use standard terminology and notation from [5]. We will assume the standard adjacency-list representation of a graph. This imposes a *numbering* from 1 to n on the vertices.

A graph $G = (V, E)$ is a *probe graph* if the vertex set is partitioned into P, the set of probes, and N the set of non-probes. In this case, every edge of E is adjacent to at least one probe. We denote this by $G = (P, N, E)$. If X is a nonempty subset of V, let $G[X]$ denote the subgraph of G induced by X, together with the classification of members of X as probes or non-probes.

Let $N(v)$ denote the *open neighborhood* of v, that is, the set of neighbors of v in G, and let $N[v]$ denote its *closed neighborhood*, that is, $\{v\} \cup N(v)$.

An *interval model* of an *interval graph* is a set of intervals, one for each vertex, such that two vertices are adjacent if and only if their intervals intersect. Without loss of generality, we assume that no two endpoints of intervals coincide. Similarly, an *interval model* of a probe interval graph is a set of intervals, one for each vertex, such that two vertices are adjacent if and only if their intervals intersect *and at least one of the vertices is a probe.* If R is an interval model of a (probe) interval graph G and X is a nonempty subset of V, let $R[X]$ denote the set of intervals of members of X. Note that $R[X]$ is an interval model of $G[X]$.

We define the *cliques* of a graph to be its *maximal* complete subgraphs. In an interval model R of a graph G, each clique of G corresponds to the set of vertices whose intervals intersect a unique *clique segment* in R. A clique segment occurs where a right endpoint is immediately to the right of a left endpoint.

The *clique matrix* of a graph is a 0-1 matrix that has one column for each clique, one row for each vertex, and a 1 in row i, column j if and only if vertex i is a member of clique j. Interval graphs are exactly the set of graphs whose clique matrices have consecutive-ones orderings [6].

An interval model consists of alternating *blocks* of consecutive left endpoints and of consecutive right endpoints. The order of endpoints within a block does not change the realized graph. Therefore, we represent an interval model combinatorially by giving an ordered list of blocks, listing for each block the endpoints inside of it. In fact, a consecutive-ones ordering of the clique matrix of an interval graph is such a model, where the set of left endpoints in a column and the set of right endpoints in the column are each interpreted to be a block, where the block of left endpoints is implicitly to the left of the block of right endpoints.

A *chordal graph* is a is a graph with no induced cycle of size greater than three. Every interval graph is a chordal graph. A chordal graph has $O(V)$ cliques. It is possible to find a sparse representation of the clique matrix of a chordal graph in $O(V)$ time [15]. Booth and Lueker's algorithm [2] for recognizing interval graphs uses this to find the clique matrix or else determine that the graph is not chordal, hence not an interval graph. If it is chordal, it reduces the problem to that of finding a consecutive-ones ordering of this clique matrix.

The algorithm of [2] gives a compact representation of *all* consecutive-ones orderings of a matrix, called a *PQ-tree.* The leaves of the PQ-tree are the columns of the matrix. The PQ-tree gives all consecutive-ones orderings by constraining the orderings of children of internal nodes as follows. Some of the internal nodes are labeled *P nodes.* For such a node there is no constraint on the order of its children. Others are labeled *Q nodes.* For such a node an ordering of its children is given; the only permissible orderings of its children are the given ordering and its reverse. For a PQ-tree T, let $\Pi(T)$ denote the set of all possible orderings of its leaves, given these constraints. The algorithm of [2] either finds the PQ-tree for consecutive-ones orderings of columns of a matrix, or determines that the matrix is not a consecutive-ones matrix. Given a sparse representation of a 0-1 matrix, this takes $O(i + j + k)$ time, where i is the number of rows, j is the number of columns, and k is the number of 1's in the matrix.

Let $\Pi = \Pi(T)$. We may consider each $\pi \in \Pi$ to be a bijective function that maps elements of C, the set of columns, to elements of $\{1, 2, \ldots, |C|\}$, where for all $c \in C$, $\pi(c)$ tells the position of c in a consecutive-ones ordering represented by π. If X is a nonempty subset of C, let π_X be the bijective function that maps elements of X to $\{1, 2, \ldots, |X|\}$, giving the relative order of elements of X in π. Let $\Pi[X]$ denote $\{\pi_X | \pi \in \Pi\}$, namely, the relative orderings of elements of X given by orderings in Π. It is not hard to show that $\Pi[X]$ is the set of orderings of a PQ-tree with leaf set X; let us call this tree the *restriction* $T[X]$ *of* T *to* X. If T_1 and T_2 are two PQ-trees whose leaf sets are both C, it is not hard to show that $\Pi(T_1) \cap \Pi(T_2)$ is a set of permutations that can also be represented by a PQ-tree. Let us call this tree the *intersection* $T_1 \cap T_2$ *of* T_1 *and* T_2.

A *probe matrix* is a generalization of 0-1 matrix, which has the values $0, 1, *$, such that the $*$'s form a submatrix. The *consecutive-ones probe matrix* problem is a generalization of the consecutive-ones problem. In this problem we look for an ordering of the columns of the matrix such that there is an interpretation of the values of the $*$'s such that the 1's in every row are consecutive.

We represent a probe matrix in space proportional to the size of the matrix and the number of 1's in it, we do not represent the $*$'s explicitly. To do that, we split a probe matrix M into two submatrices. Let M_R be the submatrix of M whose rows are the rows that do not have $*$'s, and whose columns are all columns of M. Let M_C be the submatrix of M whose columns are the columns that do not have $*$'s, and whose rows are all rows of M. We represent M using sparse representations of M_R and M_C.

The rest of the paper is organized as follows. In Sect. 3 we construct a probe matrix for the input graph that generalizes the clique matrix. In Sect. 4 we construct an interval model from a consecutive-ones ordering of this matrix. In Sect. 5 we present a linear-time algorithm for the consecutive-ones probe matrix problem. Last, in Sect. 6 we determine if the interval model is unique.

3 Extension of the Clique Matrix

In this section we show how to build a probe matrix M that has the consecutive-ones property if G is a probe interval graph. The basis of this matrix is M_P, the clique matrix of $G[P]$. In addition, for every non-probe we define either new columns or a new row. A new column has a value of 0 or 1 for every row of M_P. Similarly, a new row has a value of 0 or 1 for every column of M_P. The submatrix of M induced by the new rows and the new columns consists exclusively of $*$'s, and so M is a probe matrix. We view each row of M as a *constraint*, since it limits the possible consecutive-ones orderings of M.

The graph $G[P]$ has no non-probes. Therefore, if G is a probe interval graph, then $G[P]$ is an interval graph and M_P has the consecutive-ones property. Let $x \in N$, If G is a probe interval graph then $G[P \cup \{x\}]$ is an interval graph, because a pair of non-probes is required to give rise to an interval intersection that is not an edge. This last observation is the basis of our probe matrix construction.

Therefore, we begin by finding a consecutive-ones ordering of M_P. Using [2] we can either find such an ordering or determine that G is not a probe interval graph, in $O(V + E)$ time.

Let $\mathcal{C}$ denote the set of cliques of $G[P]$. For each probe p, let $\mathcal{Q}(p)$ denote the set of cliques of $\mathcal{C}$ that contain p. In an interval model of G, the interval for p must intersect the clique segments of members of $\mathcal{Q}(p)$ and these clique segments must be consecutive. In M we represent these constraints for all P by the rows of M_P. We call these constraints *probe - clique* constraints.

Similarly, for each non-probe x, let $\mathcal{Q}(x)$ denote the set of cliques of $\mathcal{C}$ that are subsets of $N(x)$, and Q_x denote $\bigcup \mathcal{Q}(x)$. A vertex v is *simplicial* if $N(x)$ induces a complete subgraph. We split the set of non-probes into three sets: N_1 is the set of non-probes x such that $|\mathcal{Q}(x)| \geq 1$; N_2 is the set of non-simplicial vertices with $\mathcal{Q}(x) = \emptyset$; and N_3 is the set of simplicial vertices. Note that, according to this definition, a simplicial non-probe x such that $|\mathcal{Q}(x)| = 1$ is contained both in N_1 and in N_3; it does not matter into which of the two sets we put x.

The vertices of N_1 and N_2 add three kinds of rows (constraints) to M, while the vertices of N_3 add columns. We show the details below, but first we show how to partition N into these three sets. We must find for every $x \in N$ the set $\mathcal{Q}(x)$ and determine whether x is simplicial or not.

Let the *left endpoint* of a row of a consecutive-ones ordering of M_P be the column of the leftmost 1 in the row, and the *right endpoint* be the rightmost. Let $x \in N$, and assume that $G[P]$ is an interval graph. In the consecutive-ones ordering of M_P we find for every $p \in N(x)$ the left endpoint and the right endpoint of the row of p. We keep the column numbers of these two endpoints, together with their side (left or right) in a list L_x. We sort L_x for all x in linear time using a single radix sort, with x as the primary sort key, column number as the secondary sort key, and left *versus* right endpoint as the tertiary key so that if a left endpoint and right endpoint have the same primary and secondary key, the left endpoint goes to the left of the right.

We sweep through L_x from left to right, keeping a running count of the number of neighbors of x in the current column. Each time we encounter a left endpoint in L_x we increment the counter, and each time we encounter a right endpoint we decrement it. Each time we encounter a right endpoint e that follows a left endpoint, we compare the counter with the size of the clique C represented by the column of e, and include C in $\mathcal{Q}(x)$ if they are equal.

Every time we change the value of the counter, we compare it to $|N(x)|$, if these values are equal at some column C, then $N(x) \subseteq C$ and so x is simplicial.

The procedure for x takes time proportional to $|N[x]|$ for every non-probe x. Summing over all x, we have an $O(N + E)$ bound for splitting N into N_1, N_2 and N_3. We conclude with the following lemma:

Lemma 1. *In linear time we can either split N into N_1, N_2 and N_3 and find $\mathcal{Q}(x)$ for every $x \in N$, or else determine that G is not a probe interval graph.*

3.1 Non-probe - Clique Constraints

Assume that G is indeed a probe interval graph and consider the interval of $x \in N_1$ in an interval model R. The interval of x must intersect the clique segments that correspond to members of $\mathcal{Q}(x)$, and these clique segments must be consecutive in the ordering of clique segments of $G[P]$ given by any interval model of G.

The number of cliques containing a vertex v in an interval graph is bounded by $|N[v]|$, since a neighbor of v ends at the clique segment for each clique that contains v. Since $G[P \cup \{x\}]$ is an interval graph, for any $x \in N_1$, the number of cliques in $\mathcal{Q}(x)$ is bounded by $|N[x]|$.

We therefore add to M a row for each $x \in N_1$ that has 1's in the columns of $\mathcal{Q}(x)$. This adds $O(|N[x]|)$ to the size of the matrix. We call these new rows *non-probe - clique constraints.*

3.2 Non-probe - Probe Binding Constraints

The non-probe - clique constraints defined for members of N_1 are not enough. These constraints allow the interval of $x \in N_1$ to intersect the clique segments of $\mathcal{Q}(x)$ and thus intersect the intervals of Q_x, but there may be some vertices in $N(x) \setminus Q_x$. For these we add more constraints to M.

Let $x \in N_1$ and let $p \in N(x) \setminus Q_x$. Since x and p are adjacent, we know that their intervals must intersect in any model of G, and therefore $\mathcal{Q}(x) \cup \mathcal{Q}(p)$ must be consecutive. Let us call this additional constraint a *non-probe - probe binding constraint* imposed by x and p. Adding such a constraint for every such x and p will make M too large. We show that a set of new rows with a linear number of 1's is enough to enforce the non-probe - probe binding constraints.

We know that $\mathcal{Q}(x) \cap \mathcal{Q}(p) = \emptyset$, because $p \notin Q_x$. Therefore, in any interval model of G, the interval of p covers exactly one endpoint of the interval of x. Moreover, in the order of the clique segments either the rightmost member of $\mathcal{Q}(p)$ must be consecutive with the leftmost member of $\mathcal{Q}(x)$ or vice versa.

The set of probes that x is bound to is $N(x) \setminus Q_x$. In an interval model of G, we can divide this set into the set Y_1 that covers the left endpoint of x and the set Y_2 that covers the right endpoint of x. Note that although we used a specific model of G, the same Y_1 and Y_2 arise in every model (up to interchange).

Recall that the vertices are numbered arbitrarily from 1 through n. For two vertices v and u, let $v \prec u$ denote that either $\mathcal{Q}(v) \subset \mathcal{Q}(u)$ or that $\mathcal{Q}(v) = \mathcal{Q}(u)$ and v has a smaller vertex number than u does.

Since the members of Y_1 all end at the clique segment to the left of x's left endpoint and they all occupy consecutive cliques, it follows that for any two $y, y' \in Y_1$, either $\mathcal{Q}(y) \subseteq \mathcal{Q}(y')$ or $\mathcal{Q}(y') \subseteq \mathcal{Q}(y)$. It follows that Y_1 induces a linear order in the $\prec$ relation, so it has a unique a minimal member y_1 in this relation. Similarly, Y_2 has a unique minimal member y_2 in the $\prec$ relation.

By similar reasoning, each for each probe p, the $\prec$ relation on non-probes that p is bound to has at most two nonadjacent minimal members x_1 and x_2. Let us say that x and p are a *representative bound pair* if p is a minimal bound neighbor of x and x is a minimal bound neighbor of p in the $\prec$ relation.

Consider the current status of the matrix M. The matrix includes the probe - clique constraints and the non-probe - clique constraints. In $O(V+E)$ time we can either find a consecutive-ones ordering of M or determine that G is not a probe interval graph, since we cannot satisfy all the constraints. Using this ordering of M, we can determine in $O(1)$ time for two vertices $v, u \in P \cup N_1$ whether $\mathcal{Q}(v) \subseteq \mathcal{Q}(u)$ by examining the position of leftmost and rightmost 1's in the rows of u and v. Thus, the relation $\prec$ for two vertices can be determined in $O(1)$ time as well. We get that we can find the minimal bound neighbors of every vertex, and thus all the representative bound pairs, in $O(V+E)$ time.

We add to M a row for any representative pair $\{x, p\}$ that has 1's in the columns of $\mathcal{Q}(x) \cup \mathcal{Q}(p)$. This adds $O(|N[x]| + |N[p]|)$ to the size of the matrix. Since every vertex adds at most two new rows to M, the size of M remains linear in the size of G.

A consecutive-ones ordering of M satisfies the binding constraint not just for representative bound pairs, but for all such bound pairs of vertices. This is true since M already has a row with the characteristic vector of $\mathcal{Q}(v)$ for each probe or non-probe vertex v because of the probe - clique constraints and the non-probe - clique constraints.

3.3 Probe - Probe Binding Constraints

Consider $x \in N_2$. In this case, $\mathcal{Q}(x) = \emptyset$ and $N(x)$ is not a complete subgraph. If G is a probe interval graph, then in an interval model R of G, the interval of x lies between two consecutive clique segments of $R[P]$. Let C_1 and C_2 be the corresponding cliques of $G[P]$, such that C_1's segment lies to the left of C_2's. Let $Y_1 = N(x) \setminus C_2$ and let $Y_2 = N(x) \setminus C_1$. The sets Y_1 and Y_2 satisfy $Y_1 \subseteq C_1$, $Y_2 \subseteq C_2$ and $Y_1 \cap Y_2 = \emptyset$. Also, since x is not simplicial, neither Y_1 nor Y_2 is empty. Note that although we used a specific model to define Y_1 and Y_2 for x, these sets are unique for every $x \in N_2$, up to interchange between the two.

Let $y \in Y_1$ and $y' \in Y_2$. Since x is adjacent to both y and y', and does not intersect any clique segment, we know that $\mathcal{Q}(y) \cup \mathcal{Q}(y')$ must be consecutive in any interval model of G. We call this additional constraint a *probe - probe binding constraint* imposed by y and y'. As with the non-probe - probe binding constraints, we can use the same relation $\prec$ and add to M such a constraint only for a pair of minimal bound neighbors. To find these in $O(V+E)$ time, we find for each $x \in N_2$, the sets Y_1 and Y_2. All elements of Y_1 are bound to elements of Y_2, but it is enough to bind only the minimal members of the two sets to each other. This gives $O(|N_2|)$ candidate pairs for bindings. We proceed on these as in the case of probe - non-probe bindings to find representative pairs.

We add to M a row for each representative pair p, p' that has 1's in the columns of $\mathcal{Q}(p) \cup \mathcal{Q}(p')$. This adds $O(|N[p]| + |N[p']|)$ to the size of the matrix. Again, the size of M remains linear in the size of G.

3.4 Additional Segments

Last, we consider N_3. For this set of probes we do not define further constraints, but refine the probe - clique constraints. This is done by adding columns to M. As mentioned earlier, the new columns have 0 or 1 in rows of M_P and $*$ in rows that we added for N_1 and N_2.

Let $x \in N_3$, and assume that G is a probe interval graph. Let R be an interval model of G. The set $N[x]$ is a clique in G. Therefore there is a clique segment in R that is intersected by the intervals of $N[x]$.

Let $\mathcal{C}' = \{N[x] \mid x \in N_3\}$. The members of $\mathcal{C}'$ are the cliques of G that are not in $\mathcal{C}$. For each vertex v, let $\mathcal{Q}'(v)$ denote the set of members of $\mathcal{C}'$ that contain v. In an interval model of G, the interval for a probe p must intersect the clique segments that correspond to members of $\mathcal{Q}(p) \cup \mathcal{Q}'(p)$, and the clique segments of $\mathcal{Q}(p) \cup \mathcal{Q}'(p)$ must be consecutive in the left-to-right ordering of clique segments. This gives us a refinement of the probe - clique constraints.

To represent the cliques of $\mathcal{C}'$, we add a new column for every $x \in N_3$, that has a 1 in the row of a probe p if $p \in N(x)$, and 0 otherwise. Using a sparse representation of the matrix, this adds $O(|N[x]|)$ to the size of the matrix.

This concludes the construction of M. If G is a probe interval graph, then there must be a consecutive-ones ordering of the columns of M that obeys all constraints. We summarize the section in the following lemma:

Lemma 2. *It takes $O(V + E)$ time to construct the probe matrix M or else decide that G is not a probe interval graph. Moreover, if G is a probe interval graph then M is a consecutive-ones probe matrix.*

We use the algorithm of Sect. 5 to find a consecutive-ones ordering of M. If such an ordering does not exist then G is not a probe interval graph.

4 Constructing an Interval Model

In this section we use the consecutive-ones ordering of M that we found in the previous section to find an interval model of G, if one exists. The construction is similar to the construction of [2] of an interval model from the clique matrix of an interval graph. Each interval must intersect the clique segments it belongs to. In addition, realizing bindings requires some differential stretching of endpoints inside the zone between two consecutive clique segments.

Recall that we represent an interval model combinatorially by a list of alternating blocks of left and right endpoints. We begin by defining two sets of endpoints for every column C of M: C_ℓ and C_r. We show below how we populate these sets. We order the sets according to the consecutive-ones ordering of the columns of M, such that C_ℓ is to the left of C_r.

Let $v \in V \setminus N_2$. In this case, $\mathcal{Q}(v) \cup \mathcal{Q}'(v)$ is not empty. If $v \in P \cup N_1$, then $\mathcal{Q}(v) \cup \mathcal{Q}'(v)$ has a row in M. Let C be the leftmost column with 1 in this row and D be the rightmost column with 1 in this row. If $v \in N_3$, we let C and D both be the column of the clique $N[v]$. We put the left endpoint of v in C_ℓ and the right endpoint of v in D_r. Because M has a consecutive-ones ordering, this takes linear time. Let us denote the resulting interval model by R_1.

In R_1, for every column C, the segment on the line between C_ℓ and C_r is the clique segment of C. If the intervals of v and u intersect in R_1, and at least one of the two vertices is a probe, then v and u are adjacent.

However, there still might be some edges in E that are not realized by R_1. These edges are between $x \in N_1 \cup N_2$ and $p \in N(x) \setminus Q_x$. In order to realize these adjacencies we place the endpoints of vertices of N_2 and stretch the intervals of N_1, N_2 and $N(x) \setminus Q_x$ for $x \in N_1 \cup N_2$ between the clique segments of $\mathcal{C} \cup \mathcal{C}'$.

Let $x \in N_2$ and let Y_1 and Y_2 be as defined in Sect. 3.3, that is, the two sets for which x defines probe - probe constraints, such that the intervals of Y_1 are to the left of the intervals of Y_2. Let C be the rightmost column in which all rows of members of Y_1 have a 1. Let D be the leftmost column in which all rows of members of Y_2 have a 1. The columns C and D exist and are consecutive because of the probe - probe constraints. We place the left endpoint of x in D_ℓ and the right endpoint of x in C_r. Denote the construction so far by R_2. Note that R_2 is not an interval model, since we place the left endpoint of x to the right of its right endpoint. We will resolve this problem when we stretch the intervals.

The last step of the construction is to stretch intervals of vertices of N_1, N_2 and $N(x) \setminus Q_x$ for $x \in N_1 \cup N_2$. Consider two vertices v and u that are adjacent in G, but whose adjacency is not realized in R_2. Assume that v is to the left of u. (If one of them is in N_2, then it does not have a real interval, but it is still clear which one is to the left.) Because of the non-probe - probe constraints and the probe - probe constraint, we know that the set C_r, which contains the right endpoint of v, is immediately to the left of D_ℓ, which contains the left endpoint of u. We must stretch the endpoints of intervals that have unrealized intersections, between the clique segments.

For every C_r and the set to its right, D_ℓ, we split the two sets and order them as follows. We split C_r into subsets $A_0, A_1, \ldots, A_{|D_\ell|}$ and A' such that an endpoint $f \in C_r$ is in A_i if it is an endpoint of an interval of a probe p with $|N(p) \cap D_\ell| = i$, and in A' if it is an endpoint of an interval of a non-probe. Similarly we split D_ℓ into subsets $B_0, B_1, \ldots, B_{|C_r|}$ and B'. Note that some of the subsets might be empty. We replace C_r with the A_i's, where A_0 is the leftmost. We replace D_ℓ with B_i's where B_0 is the rightmost. For every endpoint $f \in B'$, we place f in a set to the right of A_j where j is the largest index such that the vertex of f is non-adjacent to all vertices of $A_0, A_1, \ldots, A_j$ and adjacent to all vertices of $A_{j+1}, A_{j+2}, \ldots, A_{|D_\ell|}$. Similarly we place every endpoint $f \in A'$ in a set on the left of the appropriate B_j. Note that the set between $A_{|D_\ell|}$ and $B_{|C_r|}$ contains both right and left endpoints. We split this set F into a set F_ℓ of left endpoints and a set F_r of right endpoints. Let us denote the resulting construction by R. See Fig. 1.

If G is a probe interval graph, then we can place every member of A' and B', and therefore we can construct R. This is because if we cannot place $f \in B'$ on the right side of any A_i or $f \in A'$ on the left side of any B_i, then any interval model of G must contain an induced chordless cycle, which is impossible.

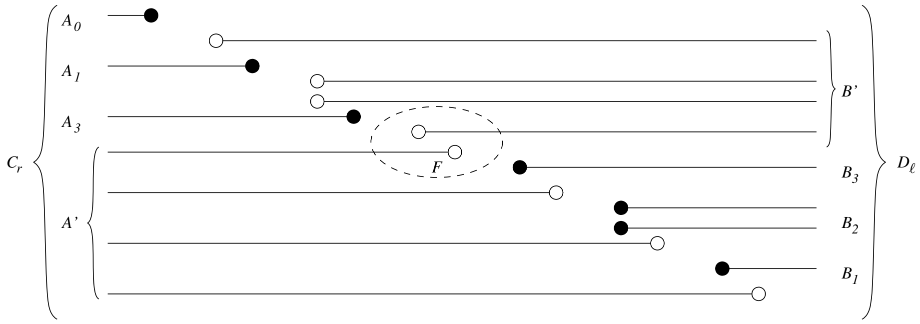

Fig. 1. Reordering the endpoints of C_r and D_ℓ. The endpoints of C_r are right endpoints and the endpoints of D_ℓ are left endpoints. Endpoints of non-probes are empty, endpoints of probes are full.

Since every interval has two endpoints, and the splitting of C_r and D_ℓ takes time proportional to the number of edges between vertices that have endpoints in these sets, in $O(V + E)$ time we can either construct R or decide that R cannot be constructed, and thus that G is not a probe interval graph.

If we manage to construct R, then it is an interval model of G. First, note that the endpoints of every vertex of N_2 are now ordered properly, that is, the left endpoint is to the left of the right endpoint. To show that R realizes G, we consider the following cases for a probe p and a vertex v, and show that their intervals in R intersect if and only if they are adjacent. If $v \in P$ then the claim is true because $R[P]$ is a model of $G[P]$. If $v \in N_1$ and $p \in Q_v$ or if $v \in N_3$ then the claim is true because the intervals intersect if and only if $(\mathcal{Q}(v) \cup \mathcal{Q}'(v)) \cap (\mathcal{Q}(p) \cup \mathcal{Q}'(p)) \neq \emptyset$. Otherwise, the claim follows by the way we stretch intervals into the region between the clique segments.

We conclude with the main theorem:

Theorem 3. *Let G be a probe graph. In $O(V + E)$ time we can construct an interval model for G, or decide that G is not a probe interval graph.*

5 Consecutive-Ones Probe Matrices

In this section we present a linear-time algorithm for the consecutive-ones probe matrix problem. Let M be a probe matrix with i rows, j columns and k 1's. We determine if M is a consecutive-ones probe matrix in $O(i + j + k)$ time. We do so by finding the PQ-tree of two 0-1 submatrices of M using [2], and combining the trees using tools of [11]. If M is a consecutive-ones probe matrix then we find a consecutive-ones ordering of it. With a modification of [11] we can find a PQ-tree that represents all consecutive-ones orderings of M. We do not present it here, because a single consecutive-ones ordering is enough, and we want to keep the description simple.

Let M_R be the submatrix of M whose rows are the rows that do not have $*$'s, and whose columns are all columns of M. Let M_C be the submatrix of M whose columns are the columns that do not have $*$'s, and whose rows are all rows of M. Let X be the set of columns of M_C.

Let T_R be the PQ-tree of M_R and let T_C be the PQ-tree of M_C. Let $T' = T_R[X] \cap T_C$. Each permutation $\sigma \in \Pi(T')$ is a permutation in both $\Pi(T_R[X])$ and in $\Pi(T_C)$. This means that $\sigma = \pi_X$ where $\pi \in \Pi(T_R)$. Assume that such a permutation $\sigma \in \Pi(T')$ exists. We can order T_R so that the relative order of leaves that are members of X is σ, because it is a restriction of some $\pi \in \Pi(T_R)$. At a P node, we order the set of children that contain leaf descendants in X according to the order of those members in σ. At a Q node, if two children contain leaf descendants in X, from the two allowed linear orders of children, we choose the one that is consistent with σ. The result is π, a consecutive-ones ordering of columns of M_R, such that $\sigma = \pi_X$ is a consecutive-ones ordering of M_C. Therefore, π is a consecutive-ones ordering of M.

On the other hand, if $\Pi(T') = \emptyset$, then there is no ordering of X that imposes a consecutive-ones ordering both for $T_R[X]$ and for T_C, and therefore M is not a probe interval matrix.

Using [2] we can find T_R and T_C in $O(i + j + k)$ time, and using [11] we can find $T_R[X]$ and from it T' in the same time bound. Choosing σ and π takes $O(j)$ time.

Theorem 4. *Let M be a probe matrix. In time linear in the size of the matrix and the number of* 1*'s in it we can either find a consecutive-ones ordering of M, or else decide that M is not a consecutive-ones probe matrix.*

6 Determining Whether a Model Is Uniquely Constrained

Recall that we represent an interval model combinatorially by a list of alternating blocks of left and right endpoints, as the order of endpoints within a block is inconsequential. Let R and R' be two interval models. We say that R and R' are *equivalent* if they are identical, or if we can get R' from R by reversing the order of its blocks and exchanging the blocks between the two endpoints of each interval. If every model of G is equivalent to R, then R is a *unique* model of G.

Let T' be as in Sect. 5, for the matrix M, found in Sect. 3, and let R be the model found in Sect. 4.

The model R is unique only if M has a unique consecutive-ones ordering up to reversal. A consecutive-ones ordering of a matrix is unique up to reversal if and only if the PQ-tree has a single internal node that is a Q node, if this is not the case for T', then R is not a unique model. The same happens also if there is more one way to produce π from σ. Specifically, this happens if there is a P node with a child that does not contain an element of X or if there is a Q node for which less than two children contain elements of X. This can be checked in time linear in the size of T'. Otherwise, the order for the columns of M is unique.

Even if M does have a unique consecutive-ones ordering, different models are possible. The matrix M defines a unique order for the cliques of G, and therefore a unique order of the clique segments in any interval model of G. Thus, for every two vertices of V such that at least one is a probe, there is a unique order defined among their endpoints. However, if there are two non-probes x and x' such that the set that contains the left endpoint of x in the model R is next to the set that contains the right endpoint of x', then we can change the order between the two endpoints. In this case as well, R is not a unique model of G. This last case can also be detected in time linear in the size of G.

References

1. Benzer, S.: On the topology of the genetic fine structure. Proc. Nat. Acad. Sci. U.S.A. 45, 1607–1620 (1959)
2. Booth, K.S., Lueker, G.S.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. J. Comput. Syst. Sci. 13, 335–379 (1976)
3. Chandler, D.B., Guo, J., Kloks, T., Niedermeier, R.: Probe matrix problems: Totally balanced matrices. In: Kao, M.-Y., Li, X.-Y. (eds.) AAIM 2007. LNCS, vol. 4508, pp. 368–377. Springer, Heidelberg (2007)
4. Chang, G.J., Kloks, T., Liu, J., Peng, S.-L.: The PIGSs full monty - a floor show of minimal separators. In: Diekert, V., Durand, B. (eds.) STACS 2005. LNCS, vol. 3404, pp. 521–532. Springer, Heidelberg (2005)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. McGraw Hill, Boston (2001)
6. Fulkerson, D.R., Gross, O.: Incidence matrices and interval graphs. Pacific J. Math. 15, 835–855 (1965)
7. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, New York (1980)
8. Golumbic, M.C.: Matrix sandwich problems. Linear Algebra and Applications 277, 239–251 (1998)
9. Golumbic, M.C., Trenk, A.N.: Tolerance Graphs. Cambridge studies in advanced mathematics 89, New York (2004)
10. Johnson, J.L., Spinrad, J.P.: A polynomial time recognition algorithm for probe interval graphs. In: SODA 2001, pp. 477–486. Association for Computing Machinery, New York (2001)
11. McConnell, R.M., de Montgolfier, F.: Algebraic operations on PQ trees and modular decomposition trees. In: Kratsch, D. (ed.) WG 2005. LNCS, vol. 3787, pp. 421–432. Springer, Heidelberg (2005)
12. McConnell, R.M.: Linear-time recognition of circular-arc graphs. Algorithmica 37, 93–147 (2003)
13. McConnell, R.M., Spinrad, J.P.: Construction of probe interval models. In: SODA 2002, pp. 866–875. Association for Computing Machinery, New York (2002)
14. McMorris, F.R., Wang, C., Zhang, P.: On probe interval graphs. Discrete Applied Mathematics 88, 315–324 (1998)
15. Rose, D., Tarjan, R.E., Lueker, G.S.: Algorithmic aspects of vertex elimination on graphs. SIAM J. Comput. 5, 266–283 (1976)
16. Uehara, R.: Canonical data structure for interval probe graphs. In: Fleischer, R., Trippen, G. (eds.) ISAAC 2004. LNCS, vol. 3341, pp. 859–870. Springer, Heidelberg (2004)
17. Zhang, P.: United states patent 5667970: Method of mapping DNA fragments (July 3, 2000)