Linear-time recognition of circular-arc graphs

Ross M. McConnell
Department of Computer Science and Engineering,
University of Colorado at Denver,
Denver, CO 80217-3364 USA,
rmcconne@carbon.cudenver.edu.

Abstract

A graph G is a **circular-arc graph** *if it is the intersection graph of a set of arcs on a circle. That is, there is one arc for each vertex of G, and two vertices are adjacent in G if the corresponding arcs intersect. We give a linear time bound for recognizing this class of graphs. When G is a member of the class, the algorithm gives a certificate in the form of a set of arcs that realize it.*

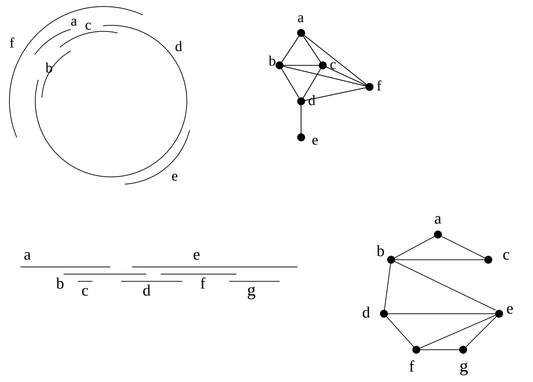


Figure 1. A circular-arc graph is the intersection graph of a set of arcs on the circle, and an interval graph is the intersection graph of a set of intervals on the line.

1. Introduction

The **intersection graph** of a family of n sets is the graph where the vertices are the sets, and the edges are the pairs of sets that intersect. Every graph is the intersection graph of some family of sets [13]. An graph is an **interval graph** if there is a way to order the universe from which the sets are drawn so that each set is consecutive. Equivalently, a graph is an interval graph if it is the intersection graph of a finite set of intervals on a line. A graph is a **circular-arc graph** if it is the intersection graph of a finite set of arcs on a circle. (See Figure 1.) An interval graph is a special case of a circular-arc graph; it is a circular-arc graph that can be represented with arcs that do not cover the entire circle.

Interval graphs and circular-arc graphs arise in scheduling problems and other combinatorial problems. Before the structure of DNA was well-understood, Seymour Benzer [1] was able to show that the set of intersections of a large number of fragments of genetic material in a virus were an interval graph. This provided strong evidence that genetic information was arranged inside a linear structure, which we now know to be true. Benzer earned the Nobel Prize partly for this work.

Being able to determine whether a graph is an interval graph or circular-arc graph constitutes **recognition** of these graph classes. However, having a representation of a graph with intervals or arcs can be helpful in solving combinatorial problems on the graph, such as isomorphism testing and finding maximum independent sets and cliques [4, 11]. Therefore, a stronger result than just recognizing the class is being able to produce the representation whenever a graph is a member of the class.

Fulkerson and Gross [8] gave an $O(n^4)$ algorithm for solving this problem on interval graphs. Booth and Lueker later improved this to linear-time [3]. Until now, a linear-time bound for circular-arc graphs has been elusive. It was initially conjectured by Booth [2] that recognition of circular-arc graphs was NP-complete. Tucker disproved this with an $O(n^3)$ algorithm [22]. Hsu improved this to $O(nm)$, where m is the number of edges [11], and Eschen and Spinrad further improved this to $O(n^2)$ [7]. We give an $O(n+m)$ bound.

One reason that the problem is harder on circular-arc graphs is that there are two ways to travel between a pair of points on a circle, and only one on a line. This introduces the need for an algorithm to make choices that are not required in the interval-graph recognition problem. Also, in an interval graph the maximal cliques correspond to regions of maximal overlap among the intervals, and there are therefore $O(n)$ maximal cliques. This plays an important role in Booth and Lueker's algorithm. The number of maximal cliques in a circular-arc graph can be exponential in n [22].

Our algorithm is based on modular decomposition, transitive orientation of comparability graphs, and algorithms on permutation graphs and interval graphs. For overviews of these topics, their applications, and their relationships to circular-arc graphs, see the books by Golumbic [10] and Roberts [20], and the survey article by Moehring [17]. A detailed presentation of the results that we give in this abstract can be found in [14].

2. Preliminaries

Let $G = (V, E)$ be a graph. Let $E' \subseteq G$ denote that $E' \subseteq E$, and $e \in G$ denote that $e \in E$. $\overline{G}$ denotes the complement of G. G^T denotes the transpose. If v is a vertex in G, $N(G, v)$ denotes the neighbors of v in G, and $N[G, v]$ denotes the **closed neighborhood**, $N(G, v) \cup \{v\}$. $N(v)$ and $N[v]$ denote these when G is understood.

We may assume without loss of generality that no endpoints of arcs coincide. Let us call the clockwise endpoint of an arc x the **left** endpoint, $l(x)$, and the counterclockwise endpoint the **right** endpoint, $r(x)$. We find it convenient to let x stand both for a vertex in G and for the arc $[l(x), r(x)]$ in the realizer. The actual positions of the endpoints on the circle are irrelevant; we consider the realizer R to be the cyclic ordering of $\{l(x) : x \in V\} \cup \{r(x) : x \in V\}$. Analogous observations apply in the special case of an interval realizer.

If G is a graph or R a realizer, and $X \subseteq V$, we let $G|X$ and $R|X$ be the subgraph or realizer induced by members of X. We treat a matrix as the adjacency matrix of a complete edge-labeled graph on V, where the members of V are numbered. The label in row i and column j of the matrix is the label of (v_i, v_j) in this graph. An unlabeled graph G can be treated as a special case; the adjacency matrix implicitly labels all edges of a complete graph with 0s and 1s. This gives an obvious generalization of many concepts from graphs to matrices. For instance, we may speak of the set V of "vertices" of a matrix and the restriction $M|X$ of the matrix to a set of vertices.

A **module of a matrix** [5, 6] corresponds to a set X of vertices such that for every vertex $y \notin X$, all directed edges in $X \times \{y\}$ have the same label, and all directed edges in $\{y\} \times X$ have the same label. In this case, y fails to **distinguish** members of X. The definition applies to unlabeled graphs as a special case [9, 10, 17]. If X and Y are disjoint modules, then every element of $X \times Y$ has the same label. A partition $\mathcal{P}$ of V is a **modular partition** if every part is a module. If P consists of one representative of each part, then $G|P$ or $M|P$ completely specify everything about the graph or matrix, except those portions of it that are internal to a single part.

Two sets A and B **overlap** if $A \cap B$, $A - B$, and $B - A$ are all nonempty. A family $\mathcal{F}$ of subsets of a set V is a **tree-decomposable family** if V and its singleton subsets are members of $\mathcal{F}$, and whenever X and Y are overlapping members of $\mathcal{F}$, then $X \cup Y$, $X \cap Y$, $X - Y$, $Y - X$, and $(X - Y) \cup (Y - X)$ are members of $\mathcal{F}$. The **strong** members of $\mathcal{F}$ are those members that properly overlap with no other member of $\mathcal{F}$. The transitive reduction of the containment relation on strong members is a tree. The nodes of the tree can be labeled **prime** and **degenerate** so that every member of $\mathcal{F}$ is either a node of the tree or a union of children of a degenerate node. This is the **decomposition tree** for the family [5, 6, 18, 19]. The modules of a graph or matrix are an example of a tree-decomposable family, and its decomposition tree is the **modular decomposition**.

An undirected graph is a special case of a digraph where each undirected edge represents two oppositely oriented directed edges. A digraph is **transitive** if, for every pair (a, b) and (b, c) of directed edges, (a, c) is also a directed edge. A transitive dag is a natural model of a partial order. A **transitive orientation** of an undirected graph is the elimination of one of the two directed edge from each undirected edge so that what remains is a transitive dag. A graph is a **comparability graph** if there exists a transitive orientation of it. Finding a transitive orientation of a comparability graph takes $O(n + m)$ time [16]. The Γ relation on an undirected graph is a relation on its directed edges where, if ab and ac are two undirected edges with common vertex a and the two other vertices b and c nonadjacent, $(a, b)\Gamma(a, c)$ and $(b, a)\Gamma(c, a)$. The Γ **implication classes** are the groups of directed edges that are directly or indirectly related to each other.

It is easily seen that any transitive orientation that contains a directed edge (a, b) must contain the implication class that contains (a, b). The implication class containing (b, a) is given by the transpose of the class containing (a, b). The union of an implication class and its transpose is a Γ **color class**.

It is easily seen that the vertices spanned by a color class are a module, and the subgraph induced by a module must contain all edges of a color class if it contains any of them. This defines a type of duality between the color classes and the modules; either of them can be used to represent the other.

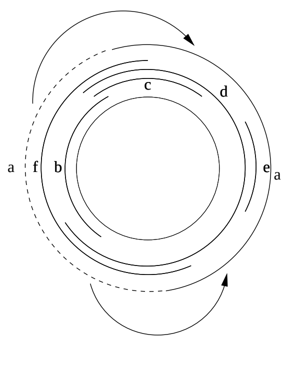
The complement of an interval graph is a comparability graph; the order of left endpoints in a realizer is a linear extension of a transitive orientation of it. A **permutation graph** is a graph G derived from two orderings of V: for $x, y \in V$, xy is an edge iff x is before y in one of the orders, and after y in the other. The two orderings are a **realizer** of the permutation graph. Reversing one of the two orderings causes the realizer to realize the complement of the graph. Both orders are linear extensions of transitive orientations of the graph.

3. Summary of the approach

Given a circular-arc realizer, let V denote the set of arcs. We can partition the edges of the complete graph on V into the following subgraphs: G_1 is the pairs of circular arcs that each contain one endpoint of the other, G_2 is the pairs that cover the circle so that each contain both endpoints of the other, G_c is the pairs where one arc contains the other, and $G_n = \overline{G}$ is non-intersecting pairs of arcs. Let D_c be the orientation of G_c where (x, y) is a directed edge iff arc x is contained in arc y.

We may think of G_1, G_2, D_c, $(D_c)^T$, and G_n as constituting an $n \times n$ adjacency matrix whose nonzero entries are labeled with the type of intersection. (In practice, we use the corresponding labeling of an adjacency-list representation.) Let us call this an **intersection matrix**. When we wish to emphasize that it is known that there is a circular-arc realizer, we will call it a **circular-arc matrix**. A circular-arc matrix is an **interval matrix** if there exists a realizer that does not cover the circle; this happens only if the corresponding graph is an interval graph. We use multiple subscripts to denote unions of these edge sets. For instance, if T is a circular-arc matrix, $G_{12c} = G_1 \cup G_2 \cup G_c = G(T)$ is the circular-arc graph it represents.

A departure of our approach from previous ones is the use of a operation on a realizer that we will call a **geometric flip**. This consists of replacing an arc x with one that has the same endpoints, but travels the opposite way around the circle. (Figure 2). This changes the types of relationships involving x's arc in a predictable way, which we will call an **algebraic flip**:



	a	b	c	d	e	f
a		t	1	2	n	c
b	c		1	2	n	c
c	1	1		c	n	1
d	2	2	t		t	2
e	n	n	n	c		n
f	t	t	1	2	n	

$\downarrow \quad \downarrow \quad \downarrow$

	a	b	c	d	e	f
a		n	1	c	t	2
b	n		1	2	n	c
c	1	1		c	n	1
d	t	2	t		t	2
e	c	n	n	c		n
f	2	t	1	2	n	

Figure 2. A flip operation has a predictable effect on a circular-arc matrix; one does not need to know a realizer of the matrix to know what effect a flip will have on it. Here, edges of G_1, G_2, G_n, D_c, and $(D_c)^T$ are labeled $1, 2, n, c$, and t, respectively.

- $(x, y) \in G_2$ becomes $(x, y) \in D_c$ and $(x, y) \in D_c$ becomes $(x, y) \in G_2$
- $(x, y) \in G_n$ becomes $(x, y) \in (D_c)^T$ and $(x, y) \in (D_c)^T$ becomes $(x, y) \in G_n$
- $(x, y) \in G_1$ remains unchanged

One matrix being obtainable from another by a sequence of algebraic flips is an equivalence relation on intersection matrices. Let us call this relation **flip-equivalence**. Every flip-equivalence class on circular-arc matrices contains an interval matrix: in a circular-arc realizer of a matrix T, if one picks a point on the circle that does not coincide with an endpoint of an arc and flips all arcs containing the point, the resulting set of arcs will fail to cover the circle at that point, and realize the interval matrix T' obtained from T by the equivalent algebraic flips.

Our algorithm produces a circular-arc realizer of G if it is a circular-arc graph. If it is not a circular-arc graph, it may return the realizer of another graph, but it is easy to check in linear time whether the returned realizer realizes G. Algorithm 3.1 summarizes the approach.

Algorithm 3.1 *Constructing a circular-arc realizer of a circular-arc graph G.*

1. *Find an intersection matrix T that is realized by some realizer of G if G is a circular-arc graph;*

2. *Perform a set of algebraic flips on T to obtain an interval matrix T';*

3. *Find an interval realizer R' of T';*

4. *Invert the flips used to obtain T' from T, but apply them as geometric flips to R'. This gives a circular-arc realizer R of T.*

4. Computing a circular-arc matrix for G

In this section, we describe the implementation of Step 1 of Algorithm 3.1. By a reduction given in [11], we may assume that G has no **duplicated neighborhoods** of the form $N[x] = N[y]$, and no **universal vertices** of the form $x : N[x] = V$.

Let $T(G)$ denote the $n \times n$ matrix T of labels that is defined as follows:

If v_i and v_j are nonadjacent then $T_{ij} = G_n$
else if $N[v_i] \subset N[v_j]$ then $T_{ij} = D_c$
else if $N[v_j] \subset N[v_i]$ then $T_{ij} = (D_c)^T$
else if $N[v_i] \cup N[v_j] = V$ and for each $v_k \in N[v_i] - N[v_j]$, $N[v_k] \subset N[v_i]$ then $T_{ij} = G_2$
else $T_{ij} = G_1$

Theorem 4.1 *[11] If G is a circular-arc graph with no duplicated neighborhoods or universal vertices, then $T(G)$ is the intersection matrix of a circular-arc realizer of G.*

A **chordal bipartite graph** is a bipartite graph with no chordless cycles of length 6 or greater. Let n_1 and n_2 denote the sizes of its bipartition classes. A **neighborhood containment test** on two vertices x and y consists of evaluating the expression $N(x) \subseteq N(y)$. A **disjoint neighborhood test** consists of evaluating whether $N(x) \cap N(y)$ is empty.

Theorem 4.2 *[21] Neighborhood containments tests between between k pairs of vertices in a chordal bipartite graph can be performed in $O(n_1 n_2 + k)$ time.*

Theorem 4.3 *[7] Let G be an arbitrary circular-arc graph, and let m_0 be a vertex whose arc contains no other in some circular-arc realizer of G. Let $D(N'[m_0], V, E_D)$ be a bipartite graph, where $N'[m_0]$ is a copy of $N[m_0]$, and $xy \in E_D$ iff $x \neq y$ and $xy \notin E$. Then D is chordal bipartite.*

In the statement of Theorem 4.3 in [7], m_0 is said to be a vertex of minimum degree, but the proof only places the restriction on m_0 that we state in Theorem 4.3.

Theorem 4.4 *Let G be a graph and let $U \subseteq V$ such that $G|U$ is an interval graph. Then it takes $O(n+m+n|V-U|)$ time to evaluate the expressions $N[x] \cap U \subseteq N[y] \cap U$ and $(N[x] \cap U) \cup (N[y] \cap U) = U$ at all pairs $\{x, y\}$ of vertices such that $x \in U$ and $y \in V - U$.*

Theorem 4.5 *It takes $O(n + m)$ time to determine that G is not a circular-arc graph, or else evaluate the expressions $N[x] \cup N[y] = V$ and $N[x] \subseteq N[y]$ and $N[y] \subseteq N[x]$ at each edge $(x, y) \in G$.*

Eschen and Spinrad give only $O(n^2)$ variants of Theorems 4.4 and 4.5, but it is quite easy to obtain the tighter bounds using their methods.

To find $T(G)$, it remains to show the following:

Theorem 4.6 *It takes $O(n + m)$ time to determine that G is not a circular-arc graph, or else evaluate the expression $N[x] \cup N[y] = V$, and for each $z \in N[x] - N[y]$, $N[z] \subseteq N[x]$.*

Eschen and Spinrad give $O(n^2)$ bounds for Theorem 4.6 only for the special case of a circular-arc graph that is coverable with two cliques. We need a linear time bound for arbitrary circular-arc graphs. We outline the proof in the remainder of this section.

Theorem 4.7 *Let H be a graph, and let $\{A, B, C\}$ be a partition of its vertices such that $H|(B \cup C)$ is an interval graph, but the edges of $H|B$ are unknown. It takes $O(m + n|A \cup B|)$ time to evaluate $N[x] \cap C \subseteq N[y] \cap C$ at all adjacent pairs $\{x, y\}$ such that at most one of x and y is in $A \cup B$.*

Using tricks from [7], it is easy to solve this problem if an interval model of a graph H' on $B \cup C$ is available such that $H'|C = H|C$, and edges of H' between B and C are the same as the edges of H between B and C. This is known as finding a realizer of a **probe interval graph**. Using a variant of the Δ tree from Section 5, below, we are able to solve this subproblem within the required time bound.

Flipping of arcs now provides the required reductions for the proof of Theorem 4.6. We again assume that G has no duplicated neighborhoods or universal vertices. Flipping y causes it to lose precisely those neighbors whose arcs, hence whose neighborhoods, are contained in $N[y]$. If $N[x] \cup N[y] = V$ then flipping y and testing whether its neighborhood is now contained in $N[x]$ tells whether for every $z \in N[y] - N[x]$, $N[z] \subseteq N[y]$.

We take advantage of this observation by computing part of a graph H that would result from adding copies of a set F of arcs in $T(G)$ that may be involved in G_2

relationships, flipping these copies, and testing neighborhood containments involving them. In particular, whenever x and y are adjacent, $N[x] \cup N[y] = V$, and the degree of x in G is at least the degree of y in G, we make sure that there is a copy x' of x in F. Since $deg(x) = \Omega(n)$, the size of H is still $O(n + m)$ after members of F are flipped. However, if $x', y' \in F$, we cannot yet tell whether x' and y' are adjacent in H, as this depends on whether $xy \in G_1$ or $xy \in G_2$, so $H|F$ remains unknown.

Let m_0 be a vertex of degree $O(m/n)$ whose arc is contained in no other in $T(G)$. Let $X = N[H, m_0] \cap V$, $Y = N[H, m_0] \cap F$. Define bipartite graphs $D_1(X \cup Y, V)$ and $D_2(X, V \cup F)$ as described in Theorem 4.3. The edges of H that are required to compute each of these are known. Each must be chordal bipartite, as they are induced subgraphs of the unknown chordal bipartite graph on H defined by Theorem 4.3. Note also that $H_I = H|((V - X) \cup (F - Y))$ is an interval graph, since in any realizer of H, the corresponding arcs fail to cover m_0.

Break the problem into two parts: $N[H, y'] \cap X \subseteq N[H, x] \cap X$ and $N[H, y] \cap (V - X) \subseteq N[H, x] \cap (V - X)$. The cases depend on whether $x \in X$ or $x \in V - X$, and whether $y' \in Y$ or $y' \in F - Y$. In each case, the time bound follows from the fact that $|F|$ and $|N[m_0]|$ are $O(m/n)$. The technique for most cases is illustrated by the case where $x \in V - X, y' \in Y$: For $N[H, y'] \cap (V - X) \subseteq N[H, x] \cap (V - X)$, apply Theorem 4.4, and for $N[H, y'] \cap X \subseteq N[H, x] \cap X$, apply Theorem 4.2 to D_2.

The outlying case is finding $N[H, y'] \cap (V - X) \subseteq N[H, x] \cap (V - X)$ when $x \in X$ and $y' \in F - Y$. Let H_P be the probe interval graph obtained by omitting the unknown edges of $H|(F - Y)$ from H_I. Apply Theorem 4.7 on $H|(V \cup (F - Y))$, using $A = X$, $B = (F - Y)$, and $C = (V - X)$.

5. Finding an interval realizer of an interval matrix

The left-endpoint order in an interval realizer R of an interval matrix T is a linear extension of a transitive orientation D_{1n} of G_{1n} in the matrix. Let us call D_{1n} an *interval orientation*; D_{1n} is uniquely defined by the left endpoint order (or, symmetrically, by the right endpoint order). Conversely, the left endpoint order is given by $(D_c)^T \cup D_{1n}$ and the right endpoint order is given by $D_c \cup D_{1n}$.

From this observation, it is easy to get an interval realizer in linear time, given a linear extension of D_{1n}. The trick is given as part of the permutation-graph recognition algorithm of [16]; G_c and G_{1n} are complementary comparability graphs, hence permutation graphs, and the left endpoint order and right endpoint order are the two permutations that realize them. This reduces the problem of finding an interval realizer to the problem of finding a linear extension of an interval orientation of G_{1n}.

The transitive orientation algorithm of [16] can be used to find a linear extension of a transitive orientation of G_{1n} in linear time, but this is not sufficient to ensure an interval orientation of G_{1n}. Interval orientations place additional constraints on the solution. For instance, the restriction of an interval orientation to G_n must also be a transitive orientation of G_n, and not all transitive orientations of G_{1n} have this property.

5.1. A Γ-like relation for interval orientations

Let us define an analog Δ of the Γ relation for the problem of finding an interval orientation instead of just a transitive orientation. (The Γ relation involves two edges joined at one end, resembling the letter Γ, while Δ involves three relationships that make a triangle.) Let Γ_n denote the Γ relation on G_n, let Γ_1 denote the Γ relation on G_1, and let Γ_{1n} denote the Γ relation on G_{1n}.

Definition 5.1 *Let $\{a, b, c\}$ be three vertices. Then $(a, b)\Delta(a, c)$ and $(b, a)\Delta(c, a)$ if one of the following applies:*

- *$(a, b)\Gamma_n(a, c)$ (i.e. $ab, ac \in G_n$ and $bc \in G_{1c}$);*
- *$(a, b)\Gamma_{1n}(a, c)$ (i.e. $ab, ac \in G_{1n}$ and $bc \in G_c$);*
- *$ab \in G_n$ and $bc, ac \in G_1$.*

By analogy to Γ, we let the Δ **implication classes** be the equivalence classes of the transitive symmetric closure of Δ, and the Δ **color classes** be the union of each equivalence class and its transpose.

Definition 5.2 *Let T be an interval matrix. Let $U(T)$ denote the matrix obtained from T by replacing each instance of D_c or $(D_c)^T$ with G_c, thereby "unorienting" the directed edges in D_c. Let us say that a set M of vertices* **overlaps** *a set E' of edges of T if $T|M$ contains some members, but not all members, of E'. A module of T or of $U(T)$ is a Δ* **module** *if it is a module that is a clique of $G(T)$, or else a module X such that for no $y \in V - X$, $\{y\} \times X \subseteq G_1$.*

Theorem 5.3 *The Δ modules of $U(T)$ are a tree-decomposable family.*

Definition 5.4 *Let us call the tree decomposition of $U(T)$ the Δ* **tree** *of $U(T)$, and denote it $\Delta(U(T))$.*

Theorem 5.5 *A set of edges of G_{1n} is a Δ color class iff it is the set of edges of G_{1n} connecting children of a prime node in the Δ tree of $U(T)$ or the set of edges of G_{1n} connecting a pair of children of a degenerate node. If X and Y are two disjoint nodes of the tree, then no Δ implication class contains both directed edges in $X \times Y$ and directed edges in $Y \times X$.*

Theorem 5.6 *If T is an interval matrix, then an orientation of G_{1n} is an interval orientation iff it is a union of Δ implication classes.*

The proofs are similar to the analogous one for standard modules and transitive orientations of comparability graphs given in [10, 17].

The following is a corollary of Theorems 5.5 and 5.6:

Lemma 5.7 *Let T be an interval matrix and let R be an interval realizer of T.*

1. *If Y is a node of $\Delta(U(T))$ that is a clique of $G(T)$, the left endpoints of Y are consecutive in R and the right endpoints of Y are consecutive in R.*

2. *If Z is a node of $\Delta(U(T))$ that is not a clique, the endpoints of Z are consecutive in R.*

5.2. An algorithm for finding a realizer of an interval matrix

An interval matrix already gives the orientation D_c of G_c. To find a realizer, it remains to find the orientation D_{1n} of G_{1n}. To do this, we find a linear extension of D_{1n}; this avoids the $\Theta(n^2)$ cost of touching the edges of G_n, which are implicit in the sparse representation.

Let us say that an ordering $(X_1, X_2, ..., X_k)$ of parts of a partition of V is **consistent** with an interval orientation D_{1n} if all edges of D_{1n} that go between two parts are directed from earlier to later parts.

Let $(X_1, X_2, ..., W, ..., X_{i-1}, Y, X_{i+1}, ..., X_k)$ be an ordering of parts on vertices of T. (A requirement is that $W \neq Y$). Let $w \in W$, let $Y_n = N(G_n, w) \cap Y$, $Y_1 = N(G_1, w) \cap Y$, and $Y_c = N(G_c, w) \cap Y$. A **pivot on Y with pivot vertex** w consists of the following refinement of the partition: (X_1, X_2, ..., W, ..., X_{i-1}, Y_c, Y_1, Y_n, X_{i+1}, ..., X_k). If W is later than Y in the ordering, a symmetric operation also constitutes a pivot: (X_1, X_2, ..., X_{i-1}, Y, X_{i+1}, W, ..., X_k) becomes (X_1, X_2, ..., X_{i-1}, Y_n, Y_1, Y_c, X_{i+1}, ..., W, ..., X_k). (Notice that the order of Y_n, Y_1, Y_c, is reversed.) We assume that if Y_c, Y_1, or Y_n is empty, it is removed from the refinement. This definition is based on the pivot of [16], which divides Y into $N(w) - Y$ and $N(w) \cap Y$.

Lemma 5.8 *Let T be an interval matrix.*

1. *If a sequence of parts on vertices of T is consistent with some interval orientation D_{1n} of G_{1n} in T, then they remain consistent with D after a pivot.*

2. *Let X be the rightmost part in a sequence of parts before a pivot, let X' be the rightmost part after a pivot and let $x \in X'$. If there exists an interval orientation D_{1n} of G_{1n} where all edges of D_{1n} between $V - X and \{x\}$ are oriented toward x, then all edges of D_{1n} between $V - X'$ and $\{x\}$ are also oriented toward x.*

The proof proceeds by induction on the number of pivots, using the Δ relationships between edges of D_{1n} that already go between parts and edges of G_{1n} that go between Y_c, Y_1, and Y_n.

Suppose there are no modules of the interval matrix other than V and its singleton subsets. When one selects an arbitrary vertex x and begins with initial partition $(\{x\}, V - \{x\})$, one finishes with a linear order where the last part $\{y\}$ contains a vertex y that is a sink in an interval orientation, by Part 2. It is a source in the transpose of this orientation, which is also an interval orientation. Running pivoting a second time with $(\{y\}, V - \{y\})$, we get a linear extension of an interval orientation, by Part 1.

If the interval matrix has nontrivial modules, then pivoting will fail to break up any module that starts out in a single part. We show that during the second partition that starts with initial partition $(\{y\}, V - \{y\})$, every one of these modules is a Δ module. By Theorem 5.6, edges internal to these modules may be oriented independently of edges external to them. Thus, the Δ modules encountered can be dealt with by recursion. We get a linear bound by extending the linear-time approach of [16] to work on symmetric matrices that have nontrivial modules and $O(1)$ different kinds of entries.

6. Turning a circular-arc matrix into an interval matrix.

In this section, we describe the implementation of Step 2 of Algorithm 3.1.

We proceed by incrementally performing flips to transform an intersection matrix for the original input graph into an interval matrix. At each point, we will let T denote the current state of the matrix, and let G_c, G_n, G_1, and G_2 refer to those graphs in T. The matrix passes through the following stages:

T_0: $T_0 = T(G)$, computed as in Section 4.

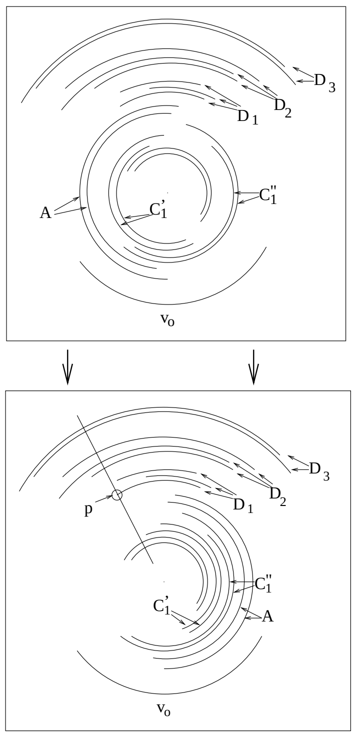


Figure 3. The set W of non-neighbors of v_0 can be divided into components $\{D_1, D_2, ..., D_k\}$ of G_{1n}, which are ordered in ascending order of containment. In T_2, the set U of neighbors of v_0 is a clique and all pairs in U are adjacent in G_c or G_1. Each component $G_c|U$ covers one endpoint of v_0's arc. In this example, $\mathcal{Q} = \{C_1', C_1'', A\}$. Flipping the members of $\mathcal{Q}$, as needed, so that they cover the right endpoint of v_0, yields T_3. At this point, it is easy to identify the arcs that contain the extreme endpoint p of D_1. Flipping these leaves a region of the circle uncovered matrix.

T_1**:** T_1 is a matrix that has a vertex v_0 of degree $O(m/n)$, such that the arc corresponding to each neighbor of v_0 contains exactly one endpoint of v_0's arc in a realizer of T_1.

Let $U = N(G(T_1), v_0)$ and $W = V - N[G(T_1), v_0]$. (It will be the case that $N(G(T_1), v_0) = N(G(T_2), v_0) = N(G(T_3), v_0)$). Let $\{D_1, D_2, ..., D_k\}$ be the components of $G_{1n}|W$. Edges between D_i and D_j are labeled G_c. These components can be ordered so that they are labeled D_c when $j > i$. The union $A(D_1)$ of arcs in D_1 is contained in the intersection of arcs in D_2, ..., D_k.

T_2**:** (See Figure 3.) T_2 shares the above properties with T_1, but $G_n|U$ and $G_2|U$ are empty, and all arcs in each component of $G_c|U$ cover a single endpoint of v_0 in any circular-arc realizer of T_2.

Let $\mathcal{Q} = \{C_1, C_2, ..., C_h\}$ denote the components of $G_c|U$ that contain an arc with an endpoint in $A(D_1)$. If two members of $\mathcal{Q}$ cover opposite endpoints of v_0, flipping all members of one of them will make them cover the same endpoint.

T_3**:** The components of $\mathcal{Q}$ are flipped, as necessary, so that they all cover the same endpoint of v_0, and they all, therefore, fail to cover one endpoint, p, of $A(D_1)$. T_3 is the resulting intersection matrix. All arcs in $V - \bigcup \mathcal{Q}$ either contain $A(D_1)$ or are disjoint from it. These can be distinguished from each other by their relationship to any member of D_1 in T_3.

T_4**:** Flipping the arcs in T_3 that contain $A(D_1)$ leaves the region of the circle adjacent to p and outside of $A(D_1)$ uncovered by any arc. T_4 is therefore an interval matrix.

Turning T_0 into T_1: We select a vertex v of minimum degree in G. If v has no G_2 relationships, we select $v_0 = v$. Otherwise, we select a neighbor of v in G_2 that is contained in no other arc according to D_c, flip it, and set v_0 to be the corresponding vertex. Now, v_0 is isolated in G_2 and is a source in D_c. We flip every neighbor in D_c; now v_0 is isolated in D_c. All arcs that intersect v_0 are neighbors in G_1, hence contain exactly one endpoint of v_0, and v_0 has at most the degree of v, which is $O(m/n)$. T_1 is the state of T at this point.

Turning T_1 into T_2: If, for a vertex $x \in U$, x contains v_0's right endpoint, then flipping x will make it contain v_0's left endpoint instead. If $x, y \in U$ and $xy \in G_n$ or $xy \in G_2$, x and y contain opposite endpoints of v_0. $G_{2n}|U$ is therefore bipartite. Flipping one bipartition class in each component of $G_{2n}|U$ eliminates all edges of $G_{2n}|U$.

Turning T_2 into T_3: (See Figure 3.) Let $\mathcal{Q}' = \{C'_1, C'_2, ..., C'_{h'}\}$ be the components of $D_c|U$ that have a member that distinguishes members of D_1, and let $\mathcal{Q}'' = \{C''_1, C''_2, ..., C''_{h''}\}$ be the set $\{C : C$ is a component of $D_c|U$, $C \notin \mathcal{Q}'$ and there is a member of D_1 that distinguishes members of $C\}$. Let $\mathcal{A}$ denote the set $\{A : A$ is a component of $G_c|U$ such that $A \times D_1 \subseteq G_1\}$. $\mathcal{Q} = \mathcal{Q}' \cup \mathcal{Q}'' \cup \mathcal{A}$.

Case A: $\mathcal{Q}''$ is nonempty: Let us consider an edge G_c to be "stronger" than an edge of G_1, and an edge G_1 to be "stronger" than an edge of G_n. For $x, y \in U$, let $x \preceq y$ denote that for every $z \in W$, the relationship of y to z is at least as strong as the relationship of x to z, and let $x \sim y$ denote that $x \preceq y$ or $y \preceq x$. If x and y cover the same endpoint of v_0, then $x \sim y$, since one of these arcs reaches at least as far into the region occupied by W, and either they both enter this region clockwise, or they both enter it counterclockwise. If A and B are two components of $G_c|U$, then let $A \preceq B$ denote that for every $x \in A$ and $y \in B$, $x \preceq y$. Let $A \sim B$ denote that $A \preceq B$ or $B \preceq A$.

Every member of each of these components covers the same endpoint of v_0. Suppose A and B cover the same endpoint. For any realizer R of T_2, $R|(A \cup B)$ is an interval realizer, since it does not cover the other endpoint of v_0. Since A and B are cliques of $G(T_2|U)$, they are Δ modules, and children of the root of $\Delta(T_2|U)$. By Lemma 5.7, we may suppose without loss of generality that all left endpoints of arcs in A precede all left endpoints of arcs in B, and that all right endpoints of arcs in A precede all right endpoints of arcs in B. Every right endpoint of B reaches at least as far into W as any arc in A, and from the same direction, hence $A \sim B$.

The following is now quite easy to show:

Lemma 6.1 *If $A \in \mathcal{Q}''$, $B \in \mathcal{Q}'' \cup \mathcal{Q}'$ and $A \neq B$, then $A \sim B$ iff A and B cover the same endpoint of v_0 in B.*

Lemma 6.1 gives a simple criterion for flipping the all members of $\mathcal{Q}$ so that they cover the same endpoint of v_0 as some $A \in \mathcal{Q}''$ does.

Case B: $\mathcal{Q}''$ is empty: Let us say that an interval x in an interval realizer R is a **leftmost** interval if its left endpoint is to the left of all others, and a **rightmost** interval if its right endpoint is to the right of all others. Recall that x distinguishes two vertices in the corresponding interval matrix if it has different relationships to them in the interval matrix.

Lemma 6.2 *Let R be an interval realizer of an interval matrix T, and let $x \in V$ that distinguishes members of $V - \{x\}$ in T. If x is leftmost in R and $G_{1n}|(V - \{x\})$ is connected, then it is not possible to realize T by removing x from R and replacing it with a rightmost interval.*

Let $c_1 \in C'_1$ such that c_1 distinguishes members of D_1. If R is a circular-arc realizer of T_2, $R|(D_1 \cup \{c_1\})$ is an interval realizer, since it fails to cover one endpoint of v_0, and c_1 is either a leftmost or rightmost interval in it. We may assume without loss of generality that R is a realizer where it is leftmost. Let D' be the interval orientation of G_{1n} given by R. Since c_1 is a leftmost interval, it is a source in $D'|(D_1 \cup \{c_1\})$, and $G_{1n}|(D_1 \cup \{c_1\})$, is connected and spans $D_1 \cup \{c_1\}$.

We may perform a sequence of pivots as in Section 5.2 in linear time, with initial partition $(\{c_1\}, D_1)$, and halting when each part in $(\{c_1\}, X_2, X_3, ..., X_k)$ is a module of T. For each i from 2 to k, let x_i denote an arbitrary member of X_i, and let $X = \{x_2, x_3, ..., x_k\}$. By Lemma 5.8 (Part 1), $(x_2, x_3, ..., x_k)$ is a linear extension of the interval orientation given by $R|X$. Since this interval orientation is forced by the Δ relation and the assumption that c_1 is leftmost, the same interval orientation applies, no matter how the representatives $\{x_2, ..., x_k\}$ are selected from $\{X_2, ..., X_k\}$. Using the interval orientation, we may construct an interval realizer R' of $R|X$. The same R' reflects $R|X$ for any selection of $\{x_2, ..., x_k\}$.

If $C'_j \neq C'_1$, then C'_j contains c_j that distinguishes members of D_1. It either distinguishes members of some X_i, or distinguishes X_i from X_j. In either case, it is easy to select $X = \{x_2, x_3, ..., x_k\}$ so that c_j distinguishes members of it. Now c_j is either leftmost or rightmost in $R|\{c_j, x_2, ..., x_k\}$, depending on whether it covers the same endpoint of v_0 as c_1 does. The outcome of this test tells whether to flip the members of C'_j in order to get them to contain the same endpoint of v_0 as C'_1 does.

Dealing with members of $\mathcal{A}$. Let T_c be the result of flipping members of $\mathcal{Q}'$ and $\mathcal{Q}''$ so that they cover the right endpoint of v_0 in some realizer R_c of T_c. With the exception of arcs in $\mathcal{A}$, this is all arcs in U that have an endpoint in $A(D_1)$.

Lemma 6.3 *There is a circular-arc realizer of T_c where all members of $\mathcal{Q}'$, $\mathcal{Q}''$, and $\mathcal{A}$ cover the right endpoint of v_0.*

Proof: Let R_c be a circular-arc realizer of T_c, Let $A \in \mathcal{A}$ be a component that does not cover the right endpoint of v_0 in R_c. A is a component of $\overline{G_1}$ and a clique. Flipping the members of A in R_c has no effects on T_c other than reversing the orientations of edges of D_c in $T_c|A$. By Lemma 5.7, these effects can be cancelled by then reversing the order of left endpoints and the order of right endpoints of members of A in R_c. $\Box$

For the analysis of the time bound, we must not assume that the input graph is a circular-arc graph. Theorem 4.5 ensures that we do not spend more than linear time finding neighborhood containments, even if G is not a circular-arc graph.

Since U has $O(m/n)$ members, the size of this array is $O(m)$. Turning T_3 into T_4 requires us to flip all members of $D_2, ..., D_k$. The intersection of all of these arcs contains D_1, from which we may deduce that each of them has $\Omega(n)$ edges in $G(T_3)$, and the number of edges of $G(T_3)$ is $O(m)$, where m is the number of edges of the input graph to the recognition problem. From this, we may spend $O(n)$ time flipping all of these and incur a total of $O(m)$. The step also requires flipping neighbors of v_0, but there are only $O(m/n)$ of these.

For turning T_2 into T_3, we may have to compute the $\preceq$ relation in U. For $x, y \in U$ in T_2, $x \preceq y$ iff $N[G_c, x] \cap W \subseteq N[G_c, y] \cap W$ and $N[x] \cap W \subseteq N[y] \cap W$ in T_2. Since $U \cup \{v_0\}$ is a clique in $G(T_2)$, $N[x] \cap W \subseteq N[y] \cap W$ iff $N[x] \subseteq N[y]$. Theorem 4.1 gives this in linear time.

To evaluate $N[G_c, x] \cap W \subseteq N[G_c, y] \cap W$ for each pair x, y of neighbors of v_0, note that if all members of $N(v_0)$ are flipped, $N(v_0)$ remains a clique, and the edges of G_n and G_c that are incident to each member of $N(v_0)$ are swapped. $N[G_c, x] \cap W \subseteq N[G_c, y] \cap W$ iff $N[x] \cap W \subseteq N[y] \cap W$ after $N(v_0)$ is flipped. We get the latter with Theorem 4.1 in linear time. Flipping $N(v_0)$ takes $O(n(m/n)) = O(m)$ time.

If $\mathcal{Q}''$ is empty, we have to perform pivots on $\{\{c_1\}, D_1\}$, but this requires linear time, as we have claimed in Section 5.2. From the resulting linear orientation, we can get the realizer for $T_3|\{x_1, x_2, ..., x_k\}$ in linear time, as explained in Section 5.2. For each $c_j \in C'_j$, it is easy to apply Lemma 6.2, and charge the cost to edges of $G(T_3)$ incident to c_j. A key observation is that we do not have to recompute the realizer for each selection of $\{x_1, x_2, ..., x_k\}$.

The remaining steps are easy to bound using elementary observations.

7. Relationship to Other Work

Tucker was the first to recognize the importance of finding neighborhood containments in recognizing circular-arc graphs [22]. This step was a bottleneck in his algorithm. Eschen and Spinrad's innovations for this step in [7] are obviously critical for our time bound. Δ trees are reminiscent of trees used by Hsu [11] for representing sets of possible arrangements of endpoints in a circular-arc realizer of a particular circular-arc matrix, rather than an interval matrix. Because the domain is more general, they are more difficult to describe, and many of the mathematical and algorithmic tools that we have developed for Δ trees are difficult to apply to them.

Johnson and Spinrad [12] have independently found an $O(n^2)$ solution to the problem mentioned in Section 4 of finding an interval realizer of a probe interval graph. They were motivated by its applications in genetics. They use a tree that is also reminiscent of Hsu's, and whose leaves are the endpoints of intervals, rather than the vertices. It represents all possible endpoint placements in a realizer of an interval graph, rather than an interval matrix. An $O(n + m \log n)$ bound for finding a realizer of a probe-interval graph is given in [15]. Additional applications and properties of Δ trees are also given there.

Acknowledgments

This research was supported by the University of Metz. The author would like to thank Dieter Kratsch for extensive inputs to this paper.

References

[1] S. Benzer. On the topology of the genetic fine structure. *Proc. Nat. Acad. Sci. U.S.A.*, 45:1607–1620, 1959.

[2] K. Booth. *PQ-Tree Algorithms.* PhD thesis, Department of Computer Science, U.C. Berkeley, 1975.

[3] S. Booth and S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.*, 13:335–379, 1976.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Algorithms.* MIT Press, Cambridge, Massachusetts, 1990.

[5] A. Ehrenfeucht and G. Rozenberg. Theory of 2-structures, part 1: Clans, basic subclasses, and morphisms. *Theoretical Computer Science*, 70:277–303, 1990.

[6] A. Ehrenfeucht and G. Rozenberg. Theory of 2-structures, part 2: Representations through labeled tree families. *Theoretical Computer Science*, 70:305–342, 1990.

[7] E. Eschen and J. Spinrad. An $O(n^2)$ algorithm for circular-arc graph recognition. *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 4:128–137, 1993.

[8] D. Fulkerson and O. Gross. Incidence matrices and interval graphs. *Pacific J. Math.*, 15:835–855, 1965.

[9] T. Gallai. Transitiv orientierbare Graphen. *Acta Math. Acad. Sci. Hungar.*, 18:25–66, 1967.

[10] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.

[11] W. Hsu. O(mn) algorithms for the recognition and isomorphism problems on circular-arc graphs. *SIAM J. Comput.*, 24:411–439, 1995.

[12] J. Johnson and J. Spinrad. A polynomial time recognition algorithm for probe interval graphs. *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, 12:477–486, 2001.

[13] E. Marczewski. Sur deux proprietés des classes d'ensembles. *Fund. Math.*, 33:303–307, 1945.

[14] R. McConnell. Linear-time recognition of circular-arc graphs. Technical Report CU-CS-914-01, University of Colorado, Boulder, 2001.

[15] R. McConnell and J. Spinrad. Construction of probe interval models. *Manuscript*.

[16] R. M. McConnell and J. P. Spinrad. Modular decomposition and transitive orientation. *Discrete Mathematics*, 201(1-3):189–241, 1999.

[17] R. H. Möhring. Algorithmic aspects of comparability graphs and interval graphs. In I. Rival, editor, *Graphs and Order*, pages 41–101. D. Reidel, Boston, 1985.

[18] R. H. Möhring. Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and boolean functions. *Annals of Operations Research*, 4:195–225, 1985.

[19] R. H. Möhring and F. J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. *Annals of Discrete Mathematics*, 19:257–356, 1984.

[20] F. S. Roberts. *Graph Theory and Its Applications to Problems of Society*. Society for Industrial and Applied Mathematics, Philadelphia, 1978.

[21] J. Spinrad. Doubly lexical ordering of dense 0-1 matrices. *Inf. Process. Lett.*, 45:229–235, 1993.

[22] A. Tucker. An efficient test for circular-arc graphs. *SIAM Journal on Computing*, 9:1–24, 1980.