SIAM J. COMPUT.
Vol. 25, No. 2, pp. 390–403, April 1996

010

LINEAR-TIME REPRESENTATION ALGORITHMS FOR PROPER CIRCULAR-ARC GRAPHS AND PROPER INTERVAL GRAPHS*

XIAOTIE DENG†, PAVOL HELL‡, AND JING HUANG§

Abstract. Our main result is a linear-time (that is, time $O(m+n)$) algorithm to recognize and represent proper circular-arc graphs. The best previous algorithm, due to A. Tucker, has time complexity $O(n^2)$. We take advantage of the fact that (among connected graphs) proper circular-arc graphs are precisely the graphs orientable as local tournaments, and we use a new characterization of local tournaments. The algorithm depends on repeated representation of portions of the input graph as proper interval graphs. Thus we also find it useful to give a new linear-time algorithm to represent proper interval graphs. This latter algorithm also depends on an orientation characterization of proper interval graphs. It is conceptually simple and does not use complex data structures. As a byproduct of the correctness proof of the algorithm, we also obtain a new proof of a characterization of proper interval graphs by forbidden subgraphs.

Key words. proper circular-arc graphs, proper interval graphs, local tournaments, representation algorithms

AMS subject classifications. 05C85, 05C75, 68Q25

1. Introduction.

An *interval graph* is the intersection graph of a family of linear intervals; a *proper interval graph* is the intersection graph of an inclusion-free family of linear intervals. A *circular-arc graph* is the intersection graph of a family of circular arcs; as above, it is called *proper* if the family can be chosen to be inclusion free. The families of intervals or of circular arcs are called *representations* of the corresponding graphs.

Algorithmic aspects of interval graphs have been extensively studied; cf. [7]. In particular, there are linear-time (i.e., time $O(m+n)$, where m and n are, respectively, the numbers of edges and of vertices of the input graph) algorithms to find a representation of a given input graph by a family of intervals if one exists [5, 14]. It is a long-standing open problem to find a linear-time algorithm for the representation of circular-arc graphs. Here we give a linear-time algorithm for the representation of *proper* circular-arc graphs. Tucker [18] gave a matrix characterization of proper circular-arc graphs and an associated representation algorithm, which runs in time $O(n^2)$ (cf. [15]).

Our algorithm depends on a linear-time method to represent proper interval graphs. Such a method can be extracted from existing algorithms for interval graphs [7, p. 195]. However, we also include in this paper a new linear-time algorithm for the representation of proper interval graphs. This is a simple incremental algorithm which does not use special data structures such as PQ-trees [5, 14]. Moreover, it is formulated in a way that makes it convenient to use as a subroutine for our main algorithm for proper circular-arc graphs. (It should be remarked that W. L. Hsu has recently announced a linear-time algorithm for the representation of general interval graphs, which does not use PQ-trees; cf. [10].)

We have given an earlier algorithm for the representation of proper circular-arc graphs with time complexity $O(\Delta m)$ in [9]; cf. also [2, 8, 13]. (Here Δ denotes the

* Received by the editors November 25, 1992; accepted for publication (in revised form) September 7, 1994. Parts of this paper have appeared, in a preliminary form, in [6].
† Department of Computing Science, York University, North York, ON M3J 1P3, Canada.
‡ School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada.
§ Department of Mathematics and Statistics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada.

maximum degree of the input graph.) Thus the present algorithm is more efficient; nevertheless, for graphs of bounded degree both algorithms are linear, and we believe the algorithm in [9] (which is conceptually much simpler) remains interesting.

Our algorithm for the representation of proper circular-arc graphs has immediate applications in situations where existing algorithms for optimization problems on proper circular-arc graphs assume that a representation by circular-arcs is given. For instance, there are $O(n)$ algorithms for solving the maximum clique problem, the maximum independent set problem, and the q-colouring problem in proper circular-arc graphs, provided a representation by circular arcs is given; cf. [4, 11, 17]. In view of our $O(m+n)$ representation algorithm, we may now conclude that all these problems for proper circular-arc graphs are solvable in time $O(m+n)$ with no further restriction.

The correctness proof of our algorithms depends on the progress we made on another problem—the understanding of the structure of local tournaments. A *local tournament* (cf. [1, 8, 13]), is an oriented graph in which the inset as well as the outset of every vertex is a tournament. Local tournaments turn out to be a useful generalization of tournaments and many interesting algorithmic problems can be efficiently solved for this class of oriented graphs (cf. [1, 8, 13]). Local tournaments are relevant to our purposes because of the fact that (among connected graphs) proper circular-arc graphs are precisely the graphs orientable as local tournaments and proper interval graphs are precisely the graphs orientable as nonstrong local tournaments. These results are essentially due to Skrien [16] (cf. also [9, 12, 13]), and they allow us to deal with orientations of the input graph instead of having to deal with representations of it. We have developed new characterizations of local tournaments and of nonstrong local tournaments, and we shall use them below to prove the correctness of our algorithms.

To capture the situations where some edges of the input graph have already been oriented and others are still undirected, we employ the notion of a *mixed graph*: it is a graph H with some directed edges (arcs) and some undirected edges such that if H contains the directed edge xy, then it contains neither the directed edge yx nor the undirected edge xy. The *inset*, respectively the *outset*, of a vertex v in a mixed graph H is the set of all vertices u in H for which uv, respectively vu, is a directed edge of H. A directed path in a mixed graph contains only directed edges (all oriented in the same direction). Note that the class of mixed graphs without directed edges is precisely the class of undirected graphs and the class of mixed graphs without undirected edges is precisely the class of oriented graphs.

Let D be a mixed graph. If xy is a directed edge of D, then we say that x dominates y and write $x \rightarrow y$; otherwise, we say that x does not dominate y and write $x \not\rightarrow y$. We say that D is *strong* if, for any two vertices x and y, D contains a directed path from x to y (and a directed path from y to x); otherwise, it is called *nonstrong*.

Let G be an undirected graph. For any vertex v, let $N(v)$ be the neighbourhood of v, i.e., the set of vertices which are adjacent to v. The *closed neighbourhood* of v is the set $N[v] = N(v) \cup \{v\}$. We define an equivalence relation on $V(G)$ in which a and b are equivalent just if $N[a] = N[b]$. The classes of this equivalence are called the *blocks* of G. It follows from the definition that two blocks are either completely adjacent (every vertex of one is adjacent to every vertex of the other) or completely nonadjacent (no vertex of one is adjacent to any vertex of the other). We shall use the terms adjacent blocks and nonadjacent blocks to describe these respective situations. If the vertices a and b of an edge ab are equivalent, we call the edge ab *balanced*; otherwise, ab is an *unbalanced* edge. We say that G is *reduced* if there are no balanced edges, i.e., if distinct vertices have distinct closed neighbourhoods.

The *underlying graph* of a mixed graph D is the undirected graph $G(D)$ with the vertex set $V(D)$ in which xy is an edge of $G(D)$ only if it is a directed or undirected edge of D. We say that D is *connected* if $G(D)$ is connected. We say that D is *reduced* if $G(D)$ is reduced. We call an arc ab of D balanced if the edge ab is balanced in $G(D)$.

2. Local tournaments. First we discuss the structure of local tournaments. A detailed treatment appears in [12] (cf. also [13]); we summarize here only the main points relevant to our algorithm. We begin with nonstrong local tournaments.

A *straight enumeration* of an oriented graph D is a linear ordering $v_1, v_2, \ldots, v_n$ of its vertices such that for each i there exist nonnegative integers k and l such that the vertex v_i has inset $\{v_{i-1}, v_{i-2}, \ldots, v_{i-k}\}$ and outset $\{v_{i+1}, v_{i+2}, \ldots, v_{i+l}\}$. Note that any arc v_iv_j has $i < j$ and the subgraph induced by $v_i, v_{i+1}, \ldots, v_j$ is a transitive tournament. An oriented graph which admits a straight enumeration is called *straight.* An undirected graph is said to have a *straight orientation* if it admits an orientation which is a straight oriented graph.

PROPOSITION 2.1. *The following properties are equivalent for an oriented graph D:*

1. *D is straight;*
2. *there exists an inclusion-free family of intervals associated with the vertices of D such that u dominates v in D if and only if the interval associated with u contains the left endpoint of the interval associated with v (the interval of u intersects the interval of v "on the left").*

Proof. 1 *implies* 2: Given a straight enumeration $v_1, v_2, \ldots v_n$ of D, we associate with v_i the interval from i to $i + d_i + 1 - \frac{1}{i}$, where d_i denotes the outdegree of v_i.

2 *implies* 1: Given an inclusion-free family of intervals, we order the vertices as $v_1, v_2, \ldots, v_n$ in such a way that the left endpoints of the corresponding intervals form an increasing sequence. □

COROLLARY 2.2. *A graph is a proper interval graph if and only if it has a straight orientation.* □

We see from the above that we can construct the intervals of a representation of G in time $O(m + n)$ provided that we have a straight enumeration of an orientation D of G. Thus our algorithm for representing a proper interval graph needs only to find a straight enumeration of an orientation of G.

Note that a straight oriented graph is a nonstrong local tournament. (In fact, it is an acyclic local tournament.) Moreover, *all* nonstrong local tournaments arise in a simple fashion from straight oriented graphs. Let S be an oriented graph and let T_v be a family of disjoint tournaments indexed by the vertices v of S. The *substitution operation* on S with respect to T_v, $v \in V(S)$ results in an oriented graph D obtained from S by replacing each v by the corresponding T_v and with each vertex of T_v dominating each vertex of T_u in D just if $v \to u$ in S. Note that each edge of each T_v is balanced and, if S is reduced, there are no other balanced edges in D. The *full reversal* of an oriented graph D is the operation of reversing the directions of all oriented edges (arcs) of D.

The following theorem describes a method to generate all possible nonstrong local tournaments (other than tournaments) as well as all possible nonstrong local-tournament orientations of a fixed undirected graph.

THEOREM 2.3 ([12, 13]). *Let D be a connected oriented graph which is not a tournament. Then D is a nonstrong local tournament if and only if it is obtained from* *a reduced straight oriented graph S (with $|V(S)| > 1$) by the substitution operation with respect to some family of tournaments $T_v, v \in V(S)$.*

Moreover, if D is a nonstrong local tournament, then for every nonstrong local-tournament orientation D' of $G(D)$, either D' or the full reversal of D' is obtained from the same S by the substitution operation with respect to a family $T'_v, v \in V(S)$, where $|T'_v| = |T_v|$ for each $v \in V(S)$. □

COROLLARY 2.4. *Let G be a connected proper interval graph and D and D' two arbitrary nonstrong local tournament orientations of G. Then D' can be obtained from D by changing the directions of some balanced arcs and then possibly performing a full reversal.* □

In order to concisely describe all possible nonstrong local tournament orientations of a fixed connected proper interval graph, we introduce a modified substitution operation in which we replace each vertex v by a complete undirected graph T_v, again with each vertex of T_v dominating each vertex of T_u if and only if $v \rightarrow u$. A *straight mixed graph* H is a mixed graph obtained from a reduced straight oriented graph S by such a substitution operation with respect to a family T_v of complete graphs. Note that the blocks of the resulting mixed graph are precisely the vertex sets of the complete graphs T_v. (In other words, each undirected edge is balanced and each directed edge is unbalanced.) Let H be a straight mixed graph. *A straight enumeration* $V_1, V_2, \ldots, V_p$ of the blocks of H is an ordering of the blocks of H in the order of the straight enumeration of the corresponding vertices of S. In the following, we shall often define a straight mixed graph by describing a straight enumeration of its blocks. Given a sequence of blocks $V_1, V_2, \ldots, V_p$ of an undirected graph G, consider the mixed graph H obtained by leaving all edges xy of G with x and y from the same V_j undirected and orienting each edge xy with $x \in V_i$, $y \in V_j$, and $i < j$ from x to y. Then this is a straight mixed graph provided that for any i and j such that V_i is adjacent to V_j, the subgraph induced by the intermediate blocks $V_i \cup V_{i+1} \cup \cdots \cup V_j$ is complete.

If we orient the undirected edges so that each T_v becomes a transitive tournament, we obtain a straight oriented graph. Conversely, starting from a straight oriented graph, we obtain a straight mixed graph by replacing all balanced arcs by the corresponding undirected edges. Therefore we may extract from Corollaries 2.2 and 2.4 the following useful result.

COROLLARY 2.5. *Each connected proper interval graph is uniquely orientable as a straight mixed graph up to full reversal.* □

A similar discussion applies to general (possibly strong) local tournaments. A *round enumeration* of an oriented graph D is a circular ordering $v_1, v_2, \ldots, v_n$ of its vertices such that for each i there exist nonnegative integers k and l such that the vertex v_i has inset $\{v_{i-1}, v_{i-2}, \ldots, v_{i-k}\}$ and outset $\{v_{i+1}, v_{i+2}, \ldots, v_{i+l}\}$; here additions and subtractions are modulo n. Note that in this case there can be arcs v_iv_j with both $i < j$ and $i > j$; however, it is still true that if v_iv_j is an arc (regardless of whether $i < j$ or $j < i$), the subgraph induced by $v_i, v_{i+1}, \ldots, v_j$ is a transitive tournament. (The subscripts are again computed modulo n.) An oriented graph which admits a round enumeration is called *round*. An undirected graph is said to have a *round orientation* if it admits an orientation which is a round oriented graph. Clearly a round oriented graph is a local tournament. It is easy to see that an induced subgraph of a round oriented graph is a round oriented graph and a straight subgraph of a straight oriented graph is a straight oriented graph. Moreover, it is clear that a straight enumeration is a special case of a round enumeration and that, in fact, a round oriented graph is straight if and only if it is nonstrong.

PROPOSITION 2.6. *The following properties are equivalent for a connected ori-*

ented graph D:

1. *D is round;*
2. *there exists an inclusion-free family of circular arcs associated with the vertices of D such that $u \to v$ in D if and only if the circular arc associated with u contains the counterclockwise endpoint of the circular arc associated with v.* □

COROLLARY 2.7. *A connected graph is a proper circular-arc graph if and only if it has a round orientation.* □

The proof of Proposition 2.6 is analogous to that of Proposition 2.1, and it also yields an $O(m+n)$ method to find a representation of a given graph G by circular arcs, provided that a round enumeration of an orientation of G is given. In fact, we may modify the interval representation given in the proof of Proposition 2.1 by identifying two vertices x and y of the real line whenever $|x-y| = n+1$. This makes the real line into a circle and the intervals into circular arcs. Thus it will be sufficient to specify a linear-time algorithm to provide a round enumeration of an oriented graph D with $G(D) = G$ if one exists and to report that one does not exist otherwise.

Just as nonstrong local tournaments are related to straight oriented graphs in Theorem 2.3, general local tournaments are related to round oriented graphs. However, the relationship is more complicated. The next theorem describes a method to generate all possible local tournaments as well as all possible local-tournament orientations of a fixed undirected graph. A *special reversal* of a round oriented graph D is defined as follows: Consider the complement $\overline{G}$ of $G(D)$. If $\overline{G}$ has an odd cycle or if $\overline{G}$ is connected, then no special reversal is possible. Otherwise ($\overline{G}$ is disconnected and bipartite), we let A be either the set of all unbalanced arcs within a fixed component of $\overline{G}$ or the set of all unbalanced arcs between two fixed components of $\overline{G}$. The special reversal corresponding to A is the operation that reverses the directions of all arcs in the set A.

THEOREM 2.8 ([12, 13]). *Let D be a connected oriented graph. Then D is a local tournament if and only if it is obtained from some reduced round oriented graph R by the substitution operation with respect to some family $T_v, v \in V(R)$, possibly followed by special reversals.*

Moreover, if D is a local tournament, then every local-tournament orientation of $G(D)$ is obtained from the same R by the substitution operation with respect to a family $T'_v, v \in V(R)$, where $|T'_v| = |T_v|$ for each $v \in V(R)$, possibly followed by special reversals and/or a full reversal. □

COROLLARY 2.8. *Let G be a connected proper circular-arc graph and D and D' be two arbitrary local-tournament orientations of G. Then D' can be obtained from D by changing the directions of some balanced arcs and then possibly performing special reversals and/or a full reversal.* □

A *round mixed graph H* is a mixed graph obtained from a reduced round oriented graph R by such a substitution operation with respect to a family T_v ($v \in V(R)$) of complete graphs. Note that if we orient the undirected edges of a round mixed graph in an arbitrary way, we obtain a local tournament, and if we do it in such a way that each T_v is a transitive tournament, then we get a round oriented graph. Conversely, if we erase the directions of all balanced arcs in a round oriented graph, we obtain a round mixed graph. Thus we obtain the following result from Corollaries 2.7 and 2.9 (using the fact that no special reversals are possible if the complement of G is connected or nonbipartite).

COROLLARY 2.9. *Let G be a connected proper circular-arc graph. If the complement of G is connected or nonbipartite, then it is uniquely orientable as a round mixed graph up to full reversal.* □

3. Proper circular-arc graphs. We now give our linear-time algorithm for the representation of proper circular arc graphs. Recall that Tucker [18] gives an $O(n^2)$ algorithm; thus we only need to deal with the case when the number of edges is small relative to n^2. Let x be a vertex of minimum degree in G. Let A be the graph induced by $N[x]$ and let $B = G - A$. We may assume that B is not a complete graph. Indeed, if B is complete, we can find a representation of G by Tucker's algorithm; since x is a vertex of minimum degree, we conclude that the number of edges of G is $m \geq \frac{n^2-2n}{4}$ and so $n^2 = O(m+n)$, i.e., Tucker's algorithm is in fact linear in this case. We may also assume that G is not a proper interval graph—this can be tested in linear time by the algorithm presented in the next section. Note that a proper circular-arc graph which is not a proper interval graph must be connected.

Thus let G be a proper circular-arc graph which is not a proper interval graph. Note that this implies that G is connected and can be oriented as a strong round oriented graph $\vec{G}$. As mentioned above, we wish to find some round orientation of G. We assume from now on that B is not complete.

PROPOSITION 3.1. *Both A and B are connected proper interval graphs.*

Proof. Let $v_1, v_2, \ldots, v_n$ be a round enumeration of $\vec{G}$ and let $x = v_1$. Let f be the least integer such that v_f dominates x in $\vec{G}$ and let g be the largest integer such that x dominates v_g in $\vec{G}$. Then $f > g$ and both the subgraph of $\vec{G}$ induced by $\{v_f, v_{f+1}, \ldots, v_1\}$ and the subgraph of $\vec{G}$ induced by $\{v_1, v_2, \ldots, v_g\}$ are transitive tournaments; hence $A = \{v_f, v_{f+1}, \ldots, v_n, v_1, v_2, \ldots, v_g\}$ and $B = \{v_{g+1}, \ldots, v_{f-1}\}$. It follows that A and B are connected graphs. Let $\vec{A}$ be the subgraph of $\vec{G}$ induced by the vertices of A and let $\vec{B}$ be defined similarly. Then both $\vec{A}$ and $\vec{B}$ are round oriented graphs. We shall argue that they are, in fact, straight oriented graphs, and hence A and B are proper interval graphs. We shall show that, in fact, both A and B are acyclic oriented graphs. The subgraphs of A induced by $\{v_f, v_{f+1}, \ldots, v_1\}$ and by $\{v_1, v_2, \ldots, v_g\}$ are transitive tournaments oriented from v_f to v_1 and from v_1 to v_g. Thus for A it only remains to verify that there is no arc $v_i v_j$ with $2 \leq i \leq g$ and $f \leq j \leq n$. Any such arc would imply that $v_i, v_{i+1}, \ldots, v_j$ is a transitive tournament, contradicting the assumption that B is not complete. This proves that $\vec{A}$ is an acyclic round oriented graph and thus a straight oriented graph. For $\vec{B}$ the proof is similar, as any arc $v_i v_j$ with $g < j < i < f$ in $\vec{B}$ would imply that v_i dominates $v_1 = x$, contradicting the definition of B. $\square$

According to the above proposition, we may construct a straight mixed orientation B' of B. For this, we may use the technique of the next section, which will also produce a straight enumeration $B_1, \ldots, B_q$ of the blocks of B'. Let L be the set of all vertices of A which are adjacent (in G) to a vertex of B_q and R the set of all vertices of A which are adjacent to a vertex of B_1. Let $C = G - L$ and $D = G - R$. Finally, let E be the subgraph of G induced by the vertices of $A \cup B_q$. Note that C and D both contain B as an induced subgraph and that E contains A as an induced subgraph.

PROPOSITION 3.2. *All three graphs C, D, and E are connected proper interval graphs.*

Proof. Consider again the round enumeration $v_1, v_2, \ldots, v_n$ of $\vec{G}$, where $x = v_1, A = \{v_f, v_{f+1}, \ldots, v_n, v_1, v_2, \ldots, v_g\}$, and $B = \{v_{g+1}, \ldots, v_{f-1}\}$. We claim that

$$C = \{v_{h+1}, v_{h+2}, \ldots, v_n, v_1, \ldots, v_{f-1}\}$$

for some $h, f \leq h \leq n$. Consider first the straight enumeration $v_{g+1}, \ldots, v_{f-1}$ of $\vec{B}$. Clearly, any block of $\vec{B}$ consists of consecutive vertices in this enumeration. Thus we obtain a straight mixed orientation of B by keeping the directions of all unbalanced edges of B according to their direction in $\vec{B}$. Since the B' is unique, up to a full reversal, we may assume, by considering a full reversal of $\vec{G}$ if necessary, that $B_q = \{v_s, v_{s+1}, \ldots, v_{f-1}\}$ for some $s, g+1 < s < f$. Now we prove that $L = \{v_f, v_{f+1}, \ldots, v_h\}$, where h is the largest integer such that v_{f-1} dominates v_h (thus $f \leq h \leq n$). Indeed, it follows from the definition of round enumeration that each $v_i, f \leq i \leq h$ is adjacent to a vertex (namely v_{f-1}) of B_q. On the other hand, if some v_j is dominated by a vertex in B_q, then it is also dominated by v_{f-1}; hence $f \leq j \leq h$ by the definition of h. Also, no v_j in A can dominate a vertex in B_q because then B would be a complete graph (recall that all vertices in B_q have the same closed neighbourhood in B). Having found L, we now know that $C = \{v_{h+1}, v_{h+2}, \ldots, v_n, v_1, \ldots, v_{f-1}\}$. In the order $v_{h+1}, v_{h+2}, \ldots, v_n, v_1, \ldots, v_{f-1}$ of vertices, there are no arcs from a later vertex to an earlier vertex because $v_{h+1}, v_{h+2}, \ldots, v_n, v_1, \ldots, v_g$ is part of the straight enumeration of $\vec{A}$, $v_{g+1}, v_{g+2}, \ldots v_{f-1}$ is our straight enumeration of $\vec{B}$, and any arc $v_i v_j$ with $v_i \in B$ and $v_j \in A, j \notin \{f, f+1, \ldots, h\}$ would imply that $v_i, v_{i+1}, \ldots, v_j$ is a transitive tournament, contradicting the definition of h. Let $\vec{C}$ be the subgraph of $\vec{G}$ induced by the vertices of C. Then $v_{h+1}, v_{h+2}, \ldots, v_n, v_1, \ldots, v_{f-1}$ is a straight enumeration of $\vec{C}$. Hence C is a proper interval graph.

The argument for D is symmetric and it also yields the fact that $R = \{v_k, v_{k+1}, \ldots, v_g\}$, where $1 < k \leq g$. In particular, we note for future reference that the sets L and R are disjoint.

Let $\vec{E}$ be the subgraph of $\vec{G}$ induced by the vertices of E. Recall that $B_q = \{v_s, v_{s+1}, \ldots, v_{f-1}\}$. We claim that $v_s, v_{s+1}, \ldots, v_n, v_1, \ldots, v_g$ is a straight enumeration of $\vec{E}$. This follows easily from the fact that $v_f, v_{f+1}, \ldots, v_n, v_1, \ldots, v_g$ is a straight enumeration of $\vec{A}$, $v_s, v_{s+1}, \ldots, v_{f-1}$ a part of our straight enumeration of $\vec{B}$, and there is no arc $v_i v_j$ with $v_i \in A$ and $v_j \in B_q$ because B is not complete. The connectivity of C, D, E is clear from the above arguments. $\square$

Now we may compute, again using techniques of the next section, straight mixed orientations C', D', E' of C, D, E, respectively. Recall that these orientations are unique up to full reversal. Our algorithm consists of combining the straight mixed graphs C', D', E' to form a round mixed orientation of G. To do this, we may have to replace some of C', D', E' by their full reversals. It must be possible to do this *consistently*, i.e., in such a way that all edges of G which are oriented in more than one of the mixed graphs C', D', E' are oriented there in the same way. To see this, consider the straight mixed graphs C'', D'', E'' obtained from $\vec{C}, \vec{D}, \vec{E}$ by replacing all balanced arcs by the corresponding undirected edges. They clearly have the property that all edges of G are oriented consistently (because they all arise from $\vec{G}$); however, according to Proposition 3.2 and Corollary 2.5, each of C', D', E' is equal to the corresponding C'', D'', E'' or its full reversal. To explicitly find the right orientations C', D', E', we may proceed as follows. Arbitrarily fix C' to be either of the two possible straight mixed orientations of C. This determines D', since C and D both contain B, which is not complete and hence contains an unbalanced arc. (In other words, fixing an orientation of C determined the direction of at least one edge of D and hence determined D'.) In turn, this determines the orientation of E since the edge $v_{f-1}v_f$ of G is an unbalanced edge of both D and E (recall that x is not adjacent to v_{f-1} but is adjacent to v_f). It is clear that these operations can be performed in time $O(m+n)$ once any straight mixed orientations of C, D, E are known. Thus assume that C', D', E' are oriented consistently and let H be the mixed orientation of G in which an edge is oriented only if it is oriented in any of the graphs C', D', E' and is oriented according to the orientation it has there; clearly, this can also be done in time $O(m+n)$.

PROPOSITION 3.3. *The mixed graph H is a round mixed graph.*

Proof. We first prove that each unbalanced edge of G is unbalanced in one of the mixed straight graphs C', D', E' (and hence it is oriented in that graph). Consider an unbalanced edge uv of G. If both u and v are vertices of B and if uv is balanced in B, then there exists a vertex $z \in A$ such that, say, z is adjacent to u but not to v. Since $L \cap R = \emptyset$ (see the last sentence of the proof of Proposition 3.2), $z \in C$ or $z \in D$. Thus u, v, z are vertices of C or D and hence the edge uv is an unbalanced edge in C or D. If uv is unbalanced in B, then it is also unbalanced in C (and D) since B is an induced subgraph of C. If both u and v are vertices of A and if uv is balanced in A, then there exists a vertex $z \in B$ such that, say, z is adjacent to u but not to v. Suppose first that z dominates u. If any vertex of B dominates u, then a vertex z' of B_q dominates u (consider the round enumeration $\ldots, z, \ldots, v_{f-2}, v_{f-1}, \ldots, u, \ldots$ of $\vec{G}$); thus $u \in L$. If z' is adjacent to v, then u, v, z are vertices of D and uv is an unbalanced edge of D; if z' is not adjacent to v, then u, v, z' are vertices of E and uv is an unbalanced edge of E. If uv is unbalanced in A, then it is also unbalanced in E. If (say) u is a vertex of A and v a vertex of B, then x is adjacent to u but not v and $u \notin L$ or $u \notin R$; thus u, v, x are vertices of C or D and uv is an unbalanced edge of C or D. From the way A', B', C', D', E' were chosen, we see that they all agree with $\vec{G}$ or the full reversal of $\vec{G}$. Therefore, H is obtained from $\vec{G}$ (or its reversal) by ignoring the directions on the balanced arcs. Therefore, H is a round mixed graph. ☐

We now summarize the algorithm.

ALGORITHM 3.4. *Let G be a graph with n vertices and m edges.*

[Step 1.] *Test if G is a proper interval graph by Algorithm* 4.4. *If it is, then represent it by linear intervals viewed as a special case of circular arcs.*

[Step 2.] *Choose a vertex x of minimum degree in G. Let A be the graph induced by $N[x]$ and let $B = G - A$. If B is a clique, then find a representation of G by Tucker's algorithm. Otherwise, proceed as follows.*

[Step 3.] *If B is not a proper interval graph, then report that G is not a proper circular-arc graph. Otherwise, orient B as a straight mixed graph B' by Algorithm* 4.4, *which also produces a straight enumeration $B_1, B_2, \ldots, B_q$ of the blocks of B'.*

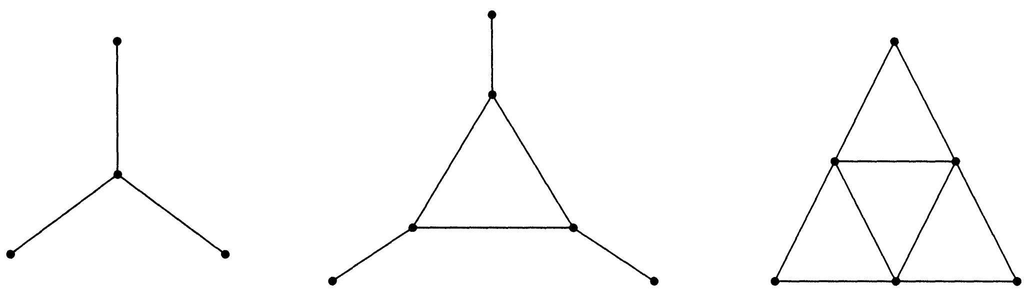
[Step 4.] *Let L be the set of vertices of A adjacent to a vertex of B_q and let R be the set of vertices of A adjacent to a vertex of B_1. Let $C = G - L$ and $D = G - R$, and let E be the subgraph of G induced by $A \cup B_q$. If any of C, D, E is not a proper interval graph, then report that G is not a proper circular-arc graph. Otherwise, orient C, D, E as straight mixed graphs C', D', E' by Algorithm* 4.4.

[Step 5.] *If it is not possible to perform full reversals on C', D', E' so that all edges are oriented consistently, then report that G is not a proper circular-arc graph. Otherwise find consistent orientations C', D', E' and construct H from G by orienting any edge uv so that $u \to v$ provided that $u \to v$ in any of the mixed graphs C', D', E'.*

[Step 6.] *Orient all remaining undirected edges of H so that each block of H becomes a transitive tournament. Let D be the resulting oriented graph.*

[Step 7.] *Transform D into a circular-arc representation of G by the method of Proposition* 2.6.

The correctness of the algorithm is implicit in our propositions. When we report that G is not a proper circular-arc graph in Step 3, then Proposition 3.1 implies that this is true. When we report this in Step 4, then Proposition 3.2 implies that this is true. When we report the same conclusion in Step 5, then this is true by the remarks following the proof of Proposition 3.2. The mixed graph H constructed in Step 5



the claw the net the tent

FIG. 1. *The claw, the net, and the tent.*

is round by Proposition 3.3. The oriented graph D constructed in Step 6 is round according to the remark following Corollary 2.8.

Steps 1, 3, and 4 are done in linear-time by Algorithm 4.4. (In Step 4, we also need to identify L and R, but this can clearly be done in time $O(m+n)$.) Step 2 also takes only linear-time according to the comments at the beginning of this section. We have already explained how the test for consistency can be done in time $O(m+n)$. The remainder of Step 5 and Step 6 are clearly linear. Step 7 can also be carried out in linear-time, as explained after Proposition 2.6.

4. Proper interval graphs. In this section, we give an $O(m+n)$-time algorithm to represent proper interval graphs. As mentioned above, such algorithms already exist. However, our algorithm is simpler than the existing algorithms, avoids difficult data structures, and directly produces the straight mixed graph orientation of the input graph that is required by Algorithm 3.4. Furthermore, the proof of correctness of our algorithm implies as a byproduct a new proof of a theorem of Wegner [19]; cf. Corollary 4.3.

Assume that G is a connected graph, as otherwise we can work separately on each component of G. Our algorithm will insert vertices of G one at a time into an already formed straight mixed graph to form a new straight mixed graph. If G is a proper interval graph, then this process continues successfully until a straight mixed graph orientation of G is obtained. This is what is needed in Algorithm 3.4. If a concrete representation by intervals is desired, then we can again orient the remaining undirected edges so that each block is a transitive tournament and then represent the resulting straight oriented graph as explained in the proof of Proposition 2.1. (All this can still be done in time $O(m+n)$.) It is easy to see that any chordless cycle of length greater than three and any of the three graphs (the claw, the net, or the tent) in Fig. 1 is not orientable as nonstrong local tournament and hence is not a proper interval graph. (This is also well known [7].) Therefore, no graph which contains a chordless cycle of length greater than three, a copy of the claw, the net, or the tent, as an induced subgraph can be a proper interval graph.

We emphasize that by Corollary 2.5 the straight mixed graph orientation of a connected proper interval graph is unique up to a full reversal. Therefore, the corresponding straight enumeration of blocks is also unique up to reversing the order of the blocks. This is crucial in what follows, even though it is not always explicitly mentioned. In particular, if we have two connected straight mixed graphs H and H' with straight enumerations $V_1, V_2, \ldots, V_p$ and $V'_1, V'_2, \ldots, V'_{p'}$ and if H is an induced subgraph of H', then we may assume (by performing a full reversal on H' if necessary) that each edge oriented both in H and H' is oriented in the same directions in both. Moreover, it is clear that two vertices equivalent in H' are also equivalent in H. Therefore, any edge unbalanced in H remains unbalanced in H'; this also means that any edge oriented in H remains oriented in H'. Suppose that H' has just one more vertex than H, say vertex v. Then each V_i breaks into two blocks $V_i \cap N(v)$ and $V_i - N(v)$ of H' (if v has both some neighbours and some nonneighbours in V_i), combines with v to form a block $V_i \cup \{v\}$ of H' (if v has exactly the same closed neighbourhood as V_i in H'), or remains a block of H' (otherwise).

Now we suppose that G is a connected proper interval graph and H is a straight mixed orientation of a connected subgraph of G. Let $V_1, V_2, \ldots, V_p$ be a fixed straight enumeration of the blocks of H. Let v be a vertex of G which is not in H but is adjacent to a vertex of H. We wish to find a straight mixed orientation H' of the subgraph of G induced by v together with the vertices of H. According to the above observations on the form of the blocks of H', it remains only to describe which edges of G not oriented in H are oriented in H'. We can do this by describing the straight enumeration of the blocks of H', because the direction of all unbalanced edges is determined by this order, to go from the earlier block towards the later block.

Before we can describe the straight enumeration of the blocks of H', we need to analyze the possible connections of v in H. In fact, these are rather restricted as far as the blocks of H are concerned.

PROPOSITION 4.1. *Let $1 \leq a < b < c \leq p$.*

1. *If v is adjacent to a vertex in V_a and to a vertex in V_c, then v is completely adjacent to V_b (i.e., v is adjacent to every vertex of V_b).*

2. *If v is adjacent to a vertex in V_b and nonadjacent to a vertex in V_a and a vertex in V_c, then V_a is nonadjacent to V_c.*

3. *If $V_a \rightarrow V_b \rightarrow V_c$ and if v is adjacent to a vertex in V_b, then v is completely adjacent to V_a or to V_c.*

Proof. Assume that v is adjacent to $x \in V_a$ and $z \in V_c$ but not to $y \in V_b$. We may assume without loss of generality that a, b, and c are chosen so that $c - a$ is minimal. Then $c - a \geq 2$ and v is not adjacent to at least one vertex in each V_d with $a < d < c$. Moreover, V_a and V_c are adjacent blocks, for otherwise any shortest path from V_a to V_c in $V_a \cup V_{a+1} \cup \cdots \cup V_c$ together with v would induce chordless cycle of length greater than three, a contradiction. (The subgraph induced by $V_a \cup V_{a+1} \cup \cdots \cup V_b$ is connected because the connectivity of H implies that each V_i is adjacent to V_{i+1}.) Since $V_1, V_2, \cdots, V_p$ is a straight enumeration of H, the vertices in $V_a \cup V_{a+1} \cup \cdots \cup V_b$ induce a complete subgraph of G. Consider the three blocks V_a, V_b, V_c; any two distinct blocks have distinct neighbourhoods in H. Thus there exists a block V_i which is adjacent to exactly one of the blocks V_a, V_b, and there exists a block V_j which is adjacent to exactly one of the blocks V_b, V_c. If V_i is adjacent to V_a but not to V_b and V_j is adjacent to V_c but not to V_b, then clearly $i < a < c < j$, and v must be completely adjacent to both V_i and V_j, as otherwise G would contain a copy of the claw centered at x or z. Now the vertices x, y, z, v together with any vertex of V_i and any of vertex of V_j induce a copy of the tent, contradicting the fact that G is a proper interval graph. Thus we may assume without loss of generality that V_i is adjacent to V_b but not to V_a. Then clearly $i > c$ and v must be completely nonadjacent to V_i, as otherwise G would contain a four-cycle induced by v, x, z and a vertex of V_i. If, additionally, V_j is adjacent to V_b but not to V_c, then we would similarly have $j < a$ and v completely adjacent to V_j. In this case G would contain a copy of the tent induced by x, y, z, v, any vertex of V_i, and any vertex of V_j. If, on the other hand, V_j is adjacent to V_c but not to V_b, then clearly $c < i < j$ and, moreover, v must be completely adjacent to V_j, as otherwise G would contain a copy of the claw centered at z. Now we conclude that G contains a chordless five-cycle induced by v, x, y, any vertex of V_i, and any vertex of V_j.

For the second statement, suppose that v is adjacent to $y \in V_b$ and is nonadjacent to $x \in V_a$ and $z \in V_c$. Assume that V_a is adjacent to V_c. The blocks V_a and V_b have distinct closed neighbourhoods in H. If there is a block V_d which is adjacent to V_b but not adjacent to V_a, then $d > c$ because V_a is adjacent to V_c. Note that v is not adjacent to any vertex in V_d according to the first statement. Hence for any $w \in V_d$, $\{x, y, w, v\}$ induces a claw in G, contradicting the fact that G is a proper interval graph. Thus there must be a block V_e which is adjacent to V_a but nonadjacent to V_b. Similarly, there is a block V_f which is adjacent to V_c but nonadjacent to V_b. Note that $e < a$ and $f > c$. Hence v is adjacent to no vertex in V_e or V_f. Choose any $u \in V_e$ and any $w \in V_f$ and note that the subgraph induced by $\{x, y, z, u, w, v\}$ is a copy of the net, again contradicting the fact that G is a proper interval graph.

For the last statement, suppose that there are three vertices $x \in V_a$, $y \in V_b$, and $z \in V_c$ such that v is adjacent to y but not to x or z. By the second statement, x is not adjacent to z. Then G contains a claw induced by $\{x, y, z, v\}$. □

Remark. We note that the proof of Proposition 4.1 applies even if we replace the assumption that G is a connected proper interval graph by the assumption that G is a connected chordal graph which contains no induced claw, net, or tent. (A graph is called *chordal* if it contains no chordless cycle of length greater than three [7].) □

Under the above assumptions that G is a connected proper interval graph (or indeed that G is a connected chordal graph which contains no induced claw, net, or tent), H a straight mixed orientation of a connected subgraph of G, and v a vertex of G but not of H, we also have the following.

PROPOSITION 4.2. *The subgraph of G induced by v and the vertices of H is a proper interval graph.*

Proof. According to part 1 of Proposition 4.1, the blocks of H that are completely adjacent to v form a (possibly empty) sequence of consecutive blocks, say, $V_l, V_{l+1}, \ldots, V_r$, for some l and r. We first assume that $1 < l < r < p$. Let $a = l - 1$ and $c = r + 1$; thus $a \geq 1$ and $c \leq p$. Note that (again by part 1 of Proposition 4.1), v is nonadjacent to all V_j with $j < a$ and $j > c$. It further follows from part 2 of Proposition 4.1 that V_a and V_c are nonadjacent. Let b be the largest integer greater than a such that V_b is adjacent to V_a, and let d be the smallest integer less than c such that V_d is adjacent to V_c. Then both $a < b < c$ and $a < d < c$. We claim that $b < d$. Suppose $b \geq d$ and let x and y be two vertices in V_a and V_c, respectively, which are nonadjacent to v; such vertices exist by the definition of l and r. Let z be any vertex of V_d. Then H has a claw induced by $\{x, y, z, v\}$.

We are now ready to describe the straight mixed graph H' in the case when $1 < l < r < p$. If the set $N(v)$ intersects V_a, then (cf. above) V_a breaks into $V_a - N(v)$ and $V_a \cap N(v)$ and we modify the straight enumeration of the blocks of H by replacing V_a with $V_a - N(v), V_a \cap N(v)$, in this order. Similarly, if the set $N(v)$ intersects V_c, then we modify the straight enumeration by replacing V_a with $V_c \cap N(v), V_c - N(v)$, in this order. Note that these operations imply that some of the previously undirected edges have become oriented, e.g., the edges oriented from $V_c - N(v)$ to $V_c \cap N(v)$. It remains to describe where the vertex v goes in the straight enumeration of H'. Recall that it can form its own block or it can be added to an existing block V_j.

If $N(v)$ intersects V_a, then we put $\{v\}$ as a separate block just after V_b. If $N(v)$ intersects V_c, then we put $\{v\}$ as a separate block just before V_d. We note that if $N(v)$ intersects both V_a and V_c, then these two instructions coincide because in this case we necessarily have $d = b + 1$. Suppose $d > b + 1$, and let x and y be vertices of $N(v) \cap V_a$ and $N(v) \cap V_c$, respectively. Let z be any vertex of V_{b+1}. Then $\{x, y, z, v\}$ forms a claw of G, a contradiction. (The vertices x, y are nonadjacent because z and y are nonadjacent and V_a precedes V_{b+1} in the straight enumeration of H_{i+1}.) It is not difficult to verify that what we have defined is indeed a straight enumeration of H'. For instance, suppose that both $N(v) \cap V_a$ and $N(v) \cap V_c$ are nonempty and consider two adjacent blocks from the straight enumeration

$$V_1, \ldots, V_{a-1}, V_a - N(v), V_a \cap N(v), \ldots, V_b, \{v\}, V_d, \ldots, V_c \cap N(v), V_c - N(v), V_{c+1}, \ldots, V_p$$

described above. If two blocks V_i and V_j with $i < j < a$ are adjacent, then the intermediate blocks are $V_i, V_{i+1}, \ldots, V_j$ and they induce a complete graph since this was the case in H; a similar argument applies if $i > j > c$. If $a < i < c$, then the intermediate blocks may additionally contain also the block $\{v\}$, which was not included in H. However, $\{v\}$ is adjacent to all blocks $V_i, V_{i+1}, \ldots V_j$ and the graph induced by the intermediate blocks is still complete. If the block $V_a - N(v)$ or the block $V_a \cap N(v)$ is adjacent to V_j, then $j \leq b$ by the definition of b and again the graph induced by the intermediate blocks is complete. If the block v is adjacent to a block V_j with, say, $j \leq b$, then it is adjacent to all blocks V_i with $j \leq i \leq b$ and once more the graph induced by the intermediate blocks is complete. A similar argument applies in the case of the adjacent blocks $\{v\}$ and $V_a \cap N(v)$. In what follows, we shall omit these simple verifications. They are all similar to the above.

If $N(v)$ intersects neither V_a nor V_c, then $N(v) = V_l \cup \cdots \cup V_r$. There may be a block V_j with $b < j < d$ adjacent to both V_l and V_r. Since by the definition of b and d we know that V_j is not adjacent to $V_{l-1} = V_a$ or $V_{r+1} = V_c$, the closed neighbourhood in H_i of any vertex of V_j is $V_l \cup \cdots \cup V_r$. Thus there can be at most one such block V_j, as distinct blocks have distinct closed neighbourhoods. Moreover, the closed neighbourhood of V_j in H' is the same as that of v. Thus we obtain a straight enumeration of H' by replacing V_j with $V_j \cup \{v\}$. On the other hand, each block V_j with $b < j < d$ is adjacent to at least one of V_l, V_r because otherwise we would have a claw formed by v and any vertices from V_l, V_j, V_r. Let u be the least integer greater than b such that V_u is adjacent to V_r and w be the largest integer smaller that d such that V_w is adjacent to V_l. Note that $b < u \leq d$ and $b \leq w < d$. We may assume that $u > w$ because otherwise there would be a block V_j with $b < j < d$ adjacent to both V_l and V_r. This means that $u = w + 1$ because each V_j with $b < j < d$ is adjacent to at least one of V_l, V_r. To obtain the desired straight enumeration of H' from $V_1, \ldots, V_p$, we insert a new block $\{v\}$ just before V_u.

The case $1 = l < r < p$ is very similar to the above arguments. We let $b = l = 1$ and $c = r + 1$ and define d as above to be the smallest integer less than c such that V_d is adjacent to V_c. Then the above discussion applies almost verbatim, except that we may have $d = b = 1$. In this case we insert the block $\{v\}$ before V_1.

Because of symmetry, it only remains to consider the following cases:

When $1 = l = r = p$, then H has only one block, V_1, and v is completely adjacent to it. Then $V_1 \cup \{v\}$ is the unique block of H'.

When $1 = l = r < p$, then $\{v\}, V_1, V_2 \cap N(v), V_2 - N(v), \ldots, V_p$ is a straight enumeration of the blocks of H'.

When $1 = l < r = p$, then v is completely adjacent to all blocks of H. We define $b = 1$ and $d = p$ and proceed as above, letting w be the largest integer greater than 1 such that V_w is adjacent to V_1 and u be the least integer smaller than p such that V_u is adjacent to V_p. Thus if $u = w$, we have the enumeration $V_1, \ldots, V_u \cup \{v\}, \ldots, V_p$ and if $u > w$, i.e., $u = w+1$ (cf. above), we have the enumeration $V_1, \ldots, V_w, \{v\}, V_u, \ldots, V_p$.

(Clearly, $u < w$ is not possible.)

The case $1 < l = r < p$ is impossible by part 3 of Proposition 4.1.

Finally, there is the trivial case when $l > r$, i.e., when v is not completely adjacent to any block of H. In the ordering the vertices of G as $v_1, \ldots, v_n$, we made sure that $v = v_{i+1}$ is adjacent to some vertex of H, say in block V_u. By part 3 of Proposition 4.1, $u = 1$ or $u = p$. Without loss of generality, we may assume that $u = 1$. If $p = 1$, i.e., if H has only one block V_1, then $\{v\}, V_1 \cap N(v), V_1 - N(v)$ is a straight enumeration of the blocks of H'. The case $p = 2$ is impossible since distinct blocks of H have distinct closed neighbourhoods, and when $p \geq 3$ then part 3 of Proposition 4.1 implies that v is nonadjacent to all V_j with $j > 1$. Then $\{v\}, V_1 \cap N(v), V_1 - N(v), V_2, \ldots, V_p$ is a straight enumeration of the blocks of H'. Now we have described the straight mixed graph H' in all cases. □

It is interesting to note that the proposition implies the following well-known result [7] (see also [3] for a refinement).

COROLLARY 4.3 ([19]). *Let G be an undirected graph. Then G is a proper interval graph if and only if it is chordal and does not contain the claw, the net, or the tent as an induced subgraph.*

Proof. We have already observed that the condition is necessary. Therefore, suppose that there is a chordal graph which does not contain an induced claw, net, or tent and which is not a proper interval graph, and let G be such a graph which has the smallest possible number of vertices. Then G must be connected; let v be a vertex of G such that $G - v$ is also connected. By the minimality of G, we know that $G - v$ can be oriented as a straight mixed graph H. Since Proposition 4.2 is valid for the graph G (cf. the sentence just before Proposition 4.2), we see that G itself is orientable as a straight mixed graph, contradicting the fact that G is not a proper interval graph. □

The proof of the above proposition is, in fact, an algorithm for inserting v into H. We summarize our algorithm for the representation of proper interval graphs as follows.

ALGORITHM 4.4. *Let G be a connected graph.*

[Step 1.] *Order the vertices of G as $v_1, v_2, \ldots, v_n$ in such a way that the subgraph induced by $\{v_1, v_2, \ldots, v_i\}$ is connected, for each $i = 1, 2, \ldots, n$. Let $v = v_1$ and $H = \{v\}$.*

[Step 2.] *Perform the following operation as long as possible: If the vertices of H are $\{v_1, v_2, \ldots, v_i\}$, then let $v = v_{i+1}$ and insert v into H as described above.*

[Step 3.] *If H does not contain all vertices, then report that G is not a proper interval graph.*

[Step 4.] *If desired, transform H into an interval representation of G by first orienting each block to be a transitive tournament and then using the method of Proposition* 2.1.

The correctness of the algorithm is assured by Proposition 4.2. Indeed, if the algorithm halts before H contains all the vertices, then some v could not be inserted and hence G is not a proper interval graph.

Our Algorithm 4.4 can be implemented in time $O(m + n)$. Since the first version of this manuscript, Corneil, Kim, Natarajan, and Olariu have published an even simpler linear-time recognition algorithm for proper interval graphs ("Simple linear time recognition of unit interval graphs," *Inform. Process. Lett.*, 55 (1995), pp. 99–104). It is possible to use their algorithm instead of ours for the purposes of Algorithm 3.4. One only has to notice that their algorithm can produce a straight mixed graph orientation of the input graph because our blocks B_i consist precisely of vertices v with equal value of their function $NextD(v) - PrevD(v)$. (Our Algorithm 4.4 remains interesting because it is incremental, and its correctness proof implies Wegner's theorem.)

Finally, we remark that we believe there is an incremental linear-time algorithm to directly compute a round mixed orientation of a given undirected graph along the lines of Algorithm 4.4. We hope to return to this idea in the future.

Acknowledgment. The authors wish to thank R. Shamir for his interest and for helpful comments.

REFERENCES

[1] J. Bang-Jensen, *Locally semicomplete digraphs: A generalization of tournaments*, J. Graph Theory, 14 (1990), pp. 371–390.

[2] J. Bang-Jensen, P. Hell, and J. Huang, *Local tournaments and proper circular arc graphs*, Technical report CSS/LCCR TR90-11, Simon Fraser University, Burnaby, BC, Canada, 1990.

[3] J. Bang-Jensen and P. Hell, *A note on chordal proper circular arc graphs*, Discrete Math., 128 (1994), pp. 395–398.

[4] B. Bhattacharya, P. Hell, and J. Huang, *A linear algorithm for maximum cliques in proper circular arc graphs*, SIAM J. Discrete Math., 9 (1996), to appear.

[5] K. S. Booth and G. S. Lueker, *Testing for the consecutive ones property using PQ-tree algorithms*, J. Comput. System Sci. 13 (1976), pp. 335–379.

[6] X. Deng, P. Hell, and J. Huang, *Recognition and representation of proper circular arc graphs*, in Integer Programming and Combinatorial Optimization, Proc. 2nd Integer Programming and Combinatorial Optimization Conference, E. Balas, G. Cornuejols, and R. Kannan, eds., Carnegie–Mellon University, Pittsburgh, PA, 1992, pp. 114–121.

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

[8] P. Hell, J. Bang-Jensen, and J. Huang, *Local tournaments and proper circular arc graphs*, in Algorithms, Springer-Verlag Lecture Notes in Computer Science, T. Asano, T. Ibaraki, H. Imai, and T. Nishizeki, eds., Springer-Verlag, Berlin, New York, 1990, pp. 101–109.

[9] P. Hell and J. Huang, *Lexicographic orientation and representation algorithms for comparability graphs, proper circular arc graphs, and proper interval graphs*, J. Graph Theory, 20 (1995), pp. 361–374.

[10] W. L. Hsu, *A simple test of interval graphs*, in Proc. SIAM Conference on Discrete Mathematics, Vancouver, 1992.

[11] W. L. Hsu and K. H. Tsai, *Linear time algorithms on circular arc graphs*, Inform. Proc. Lett., 40 (1991), pp. 123–129.

[12] J. Huang, *On the structure of local tournaments*, J. Combin. Theory Ser. B, 63 (1995), pp. 200–221.

[13] ———, *Tournament-like oriented graphs*, Ph.D. thesis, Simon Fraser University, Burnaby, BC, Canada, 1992.

[14] N. Korte and R. H. Möhring, *An incremental linear-time algorithm for recognizing interval graphs*, SIAM J. Comput., 18 (1989), pp. 68–81.

[15] J. B. Orlin, M. A. Bonuccelli, and D. P. Bovet, *An $O(n^2)$ algorithm for coloring proper circular arc graphs*, SIAM J. Algebraic Discrete Meth., 2 (1981), pp. 88–93.

[16] D. J. Skrien, *A relationship between triangulated graphs, comparability graphs, proper interval graphs, proper circular arc graphs and nested interval graphs*, J. Graph Theory, 6 (1982), pp. 309–316.

[17] A. Teng and A. Tucker, *An $O(qn)$ algorithm to q-color a proper family of circular arcs*, Discrete Math., 55 (1985), pp. 233–243.

[18] A. Tucker, *Matrix characterization of circular arc graphs*, Pacific J. Math, 39 (1971), pp. 535–545.

[19] G. Wegner, *Eigenschaften der nerven homologische-einfactor familien in R^n*, Ph.D. thesis, Universität Gottingen, Gottingen, Germany, 1967.