Discrete Applied Mathematics 157 (2009) 3216–3230

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Certifying algorithms for recognizing proper circular-arc graphs and unit circular-arc graphs☆

Haim Kaplan, Yahav Nussbaum *

The Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel

ARTICLE INFO

Article history:
Received 16 January 2008
Received in revised form 15 February 2009
Accepted 30 July 2009
Available online 12 August 2009

Keywords:
Proper circular-arc
Unit circular-arc
Certifying algorithm
Graph recognition

ABSTRACT

We give two new linear-time algorithms, one for recognizing proper circular-arc graphs and the other for recognizing unit circular-arc graphs. Both algorithms provide either a model for the input graph, or a certificate that proves that such a model does not exist and can be authenticated in $O(n)$ time. No other previous algorithm for each of these two graph classes provides a certificate for its result.

1. Introduction

A *certifying algorithm* is an algorithm that provides a *certificate* together with its answer. A certificate is an evidence that can be used to authenticate the correctness of the answer (cf. [17,10]). An *authentication algorithm* is an algorithm that checks the validity of the certificate. Certifying algorithms reduce the risk of erroneous answer, caused by bugs in the implementation.

For example, a recognition algorithm for bipartite graphs can provide a 2-coloring of the graph as a certificate when the graph is bipartite, and an odd cycle as a certificate when the graph is not bipartite. Other graph classes that have certifying recognition algorithms include chordal graphs [24], planar graphs [17], interval graphs and permutation graphs [10], proper interval graphs [7,16] and proper interval bigraphs [7].

Given an implementation of a certifying algorithm, we have a simple way to prove that every output it provides is correct, if the authentication algorithm is correct. This is important since software is prone to errors. In addition, the ability to validate every output allows us to test an implementation of a certifying algorithm on any input, rather than limiting ourselves to a specific set of inputs with known expected outputs.

Kratsch, McConnell, Mehlhorn, and Spinrad discuss the requirements for a good certificate [10]. First, it is clear that the authentication algorithm that authenticates the certificate should be simpler than an algorithm for solving the problem itself, because otherwise, the correctness of the result may be at risk because of an error in the implementation of the authentication algorithm. Additionally, the proof of the correctness of the authentication algorithm should be easy to understand. Given a certifying algorithm with an authentication algorithm, it is enough to trust the authentication algorithm, since it proves the correctness of every output of the certifying algorithm. Moreover, if we have a correct authentication algorithm, then an implementation of the certifying algorithm can be trusted even without knowing the algorithm itself.

☆ Research was partially supported by the United States–Israel Binational Science Foundation, grant number 2006204.
* Corresponding author.
E-mail addresses: haimk@cs.tau.ac.il (H. Kaplan), yahav.nussbaum@cs.tau.ac.il (Y. Nussbaum).

0166-218X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2009.07.002

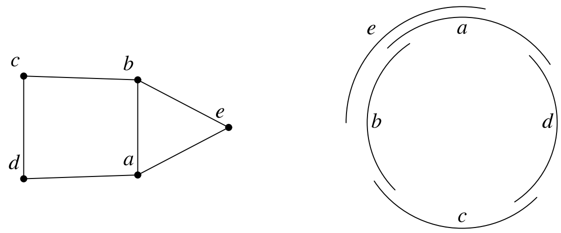


Fig. 1. A circular-arc graph and a circular-arc model of it. The model and the graph are also proper circular-arc and unit circular-arc.

The last two requirements from certificates and authentication algorithms are not formal. A formal way to measure the quality of a certificate and an authentication algorithm is the time complexity it takes to validate the certificate. A certificate is defined by [10] to be a *strong certificate*, if the time bound of its authentication algorithm is asymptotically smaller than the time bound of the best known algorithm for solving the problem itself. A certificate may be a good certificate even if it is not a strong certificate, this holds when the authentication algorithm, given the certificate, has some other advantages over solving the problem from scratch.

A *circular-arc graph* (see Fig. 1) is an intersection graph of arcs on the circle. That is, every vertex is represented by an arc, such that two vertices are adjacent if and only if the corresponding arcs intersect. The arcs constitute a *circular-arc model* of the graph. Circular-arc graphs can be recognized in linear time [14,12].

A circular-arc model in which no arc contains another arc is called a *proper circular-arc model*. A circular-arc graph that has a proper circular-arc model is a *proper circular-arc graph*. Tucker gave characterizations of proper circular-arc graphs, in terms of the adjacency matrix [22], and forbidden subgraphs [23]. Skrien [20] and Deng, Hell and Huang [4] gave characterizations that use orientation of the edges. The characterization of [4] leads to a linear-time recognition algorithm for sparse proper circular-arc graphs. For dense graphs, the linear-time algorithm of [4] uses the characterization of [22]. Spinrad [21] showed how to check the characterization of [22] in linear time, so we get a linear-time algorithm for all proper circular-arc graphs.

A circular-arc model in which all arcs are closed and of the same length is called a *unit circular-arc model*. A circular-arc graph that has a unit circular-arc model is a *unit circular-arc graph*. By definition, every unit circular-arc graph is a proper circular-arc graph. Tucker [23] gave a characterization of proper circular-arc graphs which are not unit circular-arc graphs. Recently, Durán, Gravano, McConnell, Spinrad, and Tucker [3] presented a quadratic recognition algorithm for unit circular-arc graphs, based on this characterization. This algorithm does not provide a unit circular-arc model for a unit circular-arc graph. Even more recently, Lin and Szwarcfiter [13] gave a new characterization of unit circular-arc graphs based on the length of the arcs in a proper circular-arc model. They used this characterization to derive a linear-time recognition algorithm that constructs a unit circular-arc model if the input is a unit circular-arc graph.

Note that for circular-arc models and proper circular-arc models we do not restrict the arcs to be closed (or open). Indeed, we can always perturb the arcs in a circular-arc model or a proper circular-arc model so that no two arcs share an endpoint. Therefore, it does not matter whether arcs are open or closed in such models. In contrast, for unit circular-arc models it is important that all arcs are closed. For example, consider a proper circular-arc graph which contains a cycle of length four and an independent set of size four. In a unit circular-arc model of this graph, the cycle requires that the circumference of the circle would be ≤ 4, whereas the independent set requires that it would be > 4 (see Fig. 8 below). Therefore, such a proper circular-arc graph is not a unit circular-arc graph. On the other hand, if we allow to use open arcs, we can realize the independent set using a circle of circumference 4, by four open arcs of unit length. It is known that if we restrict all arcs to be open we get the same family of graphs, and if we allow arcs to be either open or closed then every proper circular-arc graph is a unit circular-arc graph [23].

There are no previously known certifying algorithms for recognizing circular-arc graphs, proper circular-arc graphs or unit circular-arc graphs. Current algorithms construct a model of the input graph if it belongs to the appropriate graph class, but fail to provide a certificate otherwise. Moreover, even if we disregard the cost of the computation, there is no known certificate that can prove that a graph is not a circular-arc graph.

In Section 3 we present new characterizations of proper circular-arc graphs which are based on characterizations of Tucker [22,23] for this graph class and on a characterization of McConnell [15] for the consecutive-ones property. In Section 4 we present the characterizations of Tucker [23] for unit circular-arc graphs. These characterizations provide certificates for graphs that are not proper circular-arc graphs or not unit circular-arc graphs.

The two characterizations lead to linear-time certifying algorithms for recognizing proper circular-arc graphs and unit circular-arc graphs. If the input graph is a member of the graph class, then the algorithms provide an appropriate model for it. Otherwise, if the input graph is not a member of the graph class, then the algorithm provides a certificate for this answer. Prior to our work, none of these two graph classes had a certifying recognition algorithm.

If the input graph is represented by ordered adjacency lists, then the certificates that prove that the input graph is not a proper circular-arc graph or not a unit circular-arc graph can be authenticated in $O(n)$ time, where n is the number of vertices in the graph. This time bound is asymptotically better than the optimal $O(n+m)$ time bound of the recognition algorithm when the number of edges in superlinear in the number of vertices (here m denotes the number of edges). Therefore, these certificates are strong certificates.

Our algorithm for proper circular-arc graphs splits into two cases, according to whether the graph is co-bipartite or not. In each case, the algorithm is based on a different certifying algorithm for another problem.

Our algorithm for unit circular-arc graphs is based on the ideas of the recent algorithm of Durán, Gravano, McConnell, Spinrad, and Tucker [3]. We show an algorithm that runs in linear time without the complex data structures which they use, and in addition we provide a certificate for the result of the algorithm. We developed our algorithm independently of the recent linear-time algorithm of [13]. The running time of our algorithm is linear in the size of the input either when the graph is given by its set of vertices and adjacency lists, or when the graph is given by a proper circular-arc model. Note that when we are given a proper circular-arc model the size of the input is $O(n)$.

In addition to being the first certifying algorithms for proper circular-arc graphs and unit circular-arc graphs, we believe that our algorithms are also easier to implement than earlier recognition algorithms for these graph classes. This is because we do not use any complex subproblem or data structure, except of the certifying algorithm for recognizing the consecutive-ones property of McConnell [15]. If a simple algorithm is necessary and a certificate for graphs that are not proper circular-arc graphs is not required, then we can plug into our algorithm a simpler circular-ones or consecutive-ones recognition algorithm (e.g. [9]).

The structure of the rest of the paper is as follows. In Section 2 we give definitions and background. In Section 3 we describe our certifying recognition algorithm for proper circular-arc graphs. In Section 4 we describe our certifying recognition algorithm for unit circular-arc graphs. To develop this algorithm we refine some structural results about unit circular-arc graphs that may be of independent interest. We conclude in Section 5 with some suggestions for further research. An extended abstract of this paper appeared in [11].

2. Preliminaries

We consider a finite simple graph $G = (V, E)$, where $|V(G)| = n$ and $|E(G)| = m$. For a vertex v in a graph, the *(closed) neighborhood* of v, denoted by $N[v] = \{v\} \cup \{u \mid vu \in E(G)\}$ is the set of all vertices adjacent to v together with v itself. If $N[v] = V(G)$, then we call v a *universal vertex*. For $u, v \in V(G)$, if $uv \notin E(G)$ then we say that uv is a *non-edge*.

The sequence $P = (v_1, v_2, \dots v_k)$ with $v_i v_{i+1} \in E(G)$ for $i = 1, \dots, k$ is a *path*. If $v_k v_1 \in E(G)$ then P is also a *cycle*. The sequence $P = (v_1, v_2, \dots v_k)$ with $v_i v_{i+1} \notin E(G)$ for $i = 1, \dots, k$ is a *co-path*. If $v_k v_1 \notin E(G)$ then P is also a *co-cycle*. The length of a path or a co-path P, is denoted by $|P|$. A path, cycle, co-path or co-cycle in which all the vertices are distinct is *simple*.

A graph that can be partitioned into two independent sets is called *bipartite*. If G is not bipartite then it must have an odd length induced cycle. If $\overline{G}$, the complement of G, is bipartite then G is called *co-bipartite*, and its vertices are covered by two cliques.

A bipartite graph G with the bipartition (X, Y) is an *interval bigraph* if it can be represented by intervals on the line, such that the interval of $x \in X$ intersects the interval of $y \in Y$ if and only if x and y are adjacent in G. Two intervals corresponding to two vertices in X or to two vertices in Y, may or may not intersect. Müller [18] gave a polynomial time recognition algorithm for interval bigraphs.

An interval bigraph that has a model in which no arc contains another, is a *proper interval bigraph*. Hell and Huang [8] showed that the class of proper interval bigraphs, is exactly the class of the complements of co-bipartite proper circular-arc graphs. These graph classes are known to be equivalent to many other well-known graph classes including bipartite permutation graphs, bipartite AT-free graphs and bipartite trapezoid graphs (see also [2,21]). Hell and Huang [7] also gave a simple linear-time certifying algorithm for recognizing proper interval bigraphs.

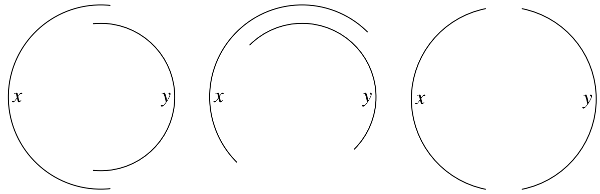
We refer to the clockwise direction of the circle as the *right* direction and to the counterclockwise direction of the circle as the *left* direction, as we view them if we stand at the center of the circle.

For a proper circular-arc graph G with a proper circular-arc model ϱ, every vertex $v \in V(G)$ has an arc in ϱ with two endpoints. We abuse the notation slightly and denote the arc of v also by v. We denote the left endpoint of the arc of v by $\ell(v)$ and the right endpoint of the arc of v by $r(v)$.

Every two arcs x and y in a proper circular-arc model ϱ either *cover the circle*, *overlap*, or are *disjoint* (see Fig. 2). Containment of arcs in a proper circular-arc model is impossible. If x overlaps y and covers $r(y)$ then we say that *x overlaps the right side of y*. Analogously, if the arc x overlaps the arc y and covers $\ell(y)$ then we say that *x overlaps the left side of y*

The *adjacency matrix* of a graph G, denoted by $M(G)$, has 1 in position (i, j) if $v_i v_j \in E(G)$, and 0 otherwise. The *augmented adjacency matrix* of G is the adjacency matrix of G with 1's on the main diagonal, that is $M^*(G) = M(G) + I$, where I is the identity matrix. We refer to the row in $M^*(G)$ that corresponds to the vertex v as row v. Likewise, we refer to the column in $M^*(G)$ that corresponds to the vertex v as column v.

A (0, 1)-matrix has the *consecutive-ones property* if its columns can be ordered so that in every row the 1's are consecutive. We can check if a matrix has the consecutive-ones property in time proportional to sum of the number of columns, the number of rows, and number of 1's in the matrix [1]. McConnell [15] gave a linear-time certifying algorithm for this property. A (0, 1)-matrix has the *circular-ones property* if its columns can be ordered so that in every row the 1's are circularly consecutive. We can check if a matrix has the circular-ones property by checking if an associated matrix has the consecutive-ones property [22] (see Section 3.2). There are also algorithms that check if a matrix has circular-ones property directly [9].



(a) Cover the circle. (b) Overlap. (c) Disjoint.

Fig. 2. Intersection types of two arcs in proper circular-arc model.

2.1. Representation

The desired graph representation for certifying algorithms on graphs is discussed in [10]. We represent a graph, as [10], by ordered adjacency lists; the edges of each list are sorted by the name of their other vertex. It is straightforward to verify that the representation of the input graph is indeed sorted in $O(m+n)$ time. This representation allows us to get the list of neighbors of a given vertex in constant time, and to give a certificate for the adjacency or non-adjacency of two vertices that can be verified in constant time. An edge is certified by its location in the ordered adjacency list. A non-edge is certified by the location of its predecessor in the adjacency list, if it would have been an edge. To collect the certificates of $O(n)$ edges and non-edges, we radix sort them, and scan the sorted list together with the adjacency lists of the graph. The running time for this sort and scan is $O(n+m)$.

If the input graph is represented by unordered adjacency lists, then the recognition algorithm remains correct and its running time remains linear. But then, the authentication algorithm would have to sort the edges of the input graph, which takes additional $O(m)$ time, using radix sort.

We represent a proper circular-arc model by a cyclic order of the endpoints of its arcs. The $2n$ endpoints in the model are indexed according to their ranks in the order, starting at any arbitrary endpoint and going to the right. Each arc is associated with the indices of its two endpoints. This representation allows both the recognition algorithm and the authentication algorithm to check for every two arcs whether they intersect or are disjoint in $O(1)$ time, using the cyclic order of their endpoints. Arcs in a unit circular-arc model also obey a length constraint, so the exact locations of the endpoints on the circle is also required.

We represent (0, 1)-matrices in a sparse way, similar to the graph representation by ordered adjacency lists. This representation allows algorithms that process matrices to run in time proportional to the sum of the number of columns, number of rows, and number of 1's in the matrix. For $M^*(G)$ the number of 1's is $O(n+m)$.

3. Certifying algorithm for recognizing proper circular-arc graphs

3.1. A characterization of proper circular-arc graphs

We define an *incompatibility graph* for proper circular-arc graphs, in a way similar to the definitions of incompatibility graphs for the consecutive-ones property [15] and for permutation graphs [10], as follows.

Let ϱ be a proper circular-arc model of G, and v_0 be a fixed vertex in G. If we start at $r(v_0)$, traverse the circle to the right, and list the vertices according to the order in which we meet their right endpoints, we get a *traversal order* $(v_0, v_1, \ldots, v_{n-1})$ of the vertices. This order defines a *traversal order relation* $R = \{(v_i, v_j) \mid i < j\}$.

Consider the traversal order of an arbitrary proper circular-arc model for G. For every $x, y \in V(G)$, (x, y) and (y, x) cannot both appear in the same traversal order relation. We say that (x, y) is *incompatible* with (y, x). For every $w \in V(G)$, the right endpoints of all the vertices in $N[w]$ must be consecutive around the circle. Assume that $v_0 \notin N[w]$. Then, in a traversal order that starts with v_0 the vertices of $N[w]$ must be consecutive. Therefore, if $x, z \in N[w]$ and $y \notin N[w]$, the vertex y cannot be between x and z. So (x, y) and (y, z) are incompatible, with w as a *witness*. Now, assume that $v_0 \in N[w]$, in this case the vertices of $N[w]$ are not necessarily consecutive in a traversal order that starts with v_0, because it might be that v_{n-1} is also in $N[w]$. But $V(G) \setminus N[w]$ must be consecutive in this traversal order, so if $x, z \notin N[w]$ and $y \in N[w]$ then (x, y) and (y, z) are incompatible, with w as a witness.

The incompatible pairs define the *incompatibility graph* $IC(G; v_0)$ of G with starting vertex v_0. The vertex set of $IC(G; v_0)$ is $\{(x, y) \mid x, y \in V(G), x \neq y\}$, which are all possible members in a traversal order relation. The edge set of $IC(G; v_0)$ consists of edges of the forms $(x, y)(y, x)$, edges of the form $(x, y)(y, z)$ such that $x, z \in N[w]$, $y \notin N[w]$ for some $w \notin N[v_0]$, and edges of the form $(x, y)(y, z)$ such that $y \in N[w]$, $x, z \notin N[w]$ for some $w \in N[v_0]$.

The definition of $IC(G; v_0)$ is analogous to the definition of the incompatibility graph for the consecutive-ones property $IC(M)$, presented by McConnell [15]. Since a consecutive-ones arrangement is linear, we do not need a starting point to define $IC(M)$. The edges of $IC(M)$ are $(x, y)(y, x)$, for every pair of columns x and y, and $(x, y)(y, z)$ such that there is a row w, with ones in the columns of x, z but not in the column of y. Actually, the incompatibility graph that we define for a proper circular-arc graph is an extension of the incompatibility graph for the consecutive-ones property to an incompatibility graph for the circular-ones property.

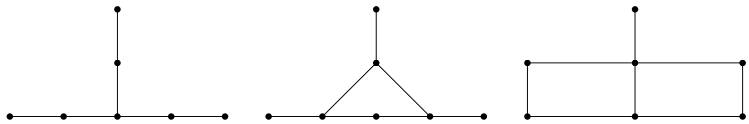


Fig. 3. Complements of forbidden subgraphs.

Theorem 1. *Let G be a proper circular-arc graph. For any $v_0 \in V(G)$, the incompatibility graph $IC(G; v_0)$ is bipartite.*

Proof. Let ϱ be a proper circular-arc model of G, let $v_0 \in V(G)$ and let R be the traversal order relation defined by the traversal order of ϱ that starts with v_0. The relation R is made of half of the vertices of $IC(G; v_0)$ since for every $x \neq y$ either $(x, y) \in R$ or $(y, x) \in R$. The relation R cannot have any incompatible pairs, so the vertices of R are an independent set in $IC(G; v_0)$. Let ϱ' be a proper circular-arc model of G that is obtained by replacing the right and left directions of ϱ. Let R' be the traversal order relation defined by the traversal order of ϱ' starting with v_0. Any vertex in $IC(G; v_0)$ that is not in R is in R'. The vertices of R' are also an independent set, and therefore $IC(G; v_0)$ is bipartite. □

To certify that G is not a proper circular-arc graph we can provide an odd cycle in one of its incompatibility graphs. We have to do so without explicitly constructing the entire incompatibility graph, since the size of this graph is at least $\Omega(n^2)$, and might be as large as $\Theta(n^4)$. Note however, that $IC(G; v_0)$ might be bipartite even when G is not a proper circular-arc graph.

Our certifying algorithm for proper circular-arc graphs consists of two cases, depending on whether G is co-bipartite or not. In the first case, when G is not co-bipartite, we use the following theorem of Tucker [22].

Theorem 2 ([22]). *Let G be a graph which is not co-bipartite. Then, G is a proper circular-arc graph if and only if $M^*(G)$ has the circular-ones property.*

To check if $M^*(G)$ has the circular-ones property, we use a reduction to the problem of checking the consecutive-ones property and use the certifying algorithm of McConnell [15] for the consecutive-ones property. If $M^*(G)$ does not have the circular-ones property, the certificate would be an odd cycle in some incompatibility graph of G.

In the second case of our algorithm, when G is co-bipartite, we use the following forbidden subgraph characterization of Tucker [23].

Theorem 3 ([23]). *Let G be a co-bipartite graph. Then, G is a proper circular-arc graph if and only if $\overline{G}$ does not contain an induced even cycle of length ≥ 6, and does not contain any of the graphs in Fig. 3 as an induced subgraph.*

Since an induced subgraph of a proper circular-arc graph is a proper circular-arc graph, any graph G such that $\overline{G}$ contains an induced even cycle of length ≥ 6, or one of the graphs in Fig. 3 as an induced subgraph, is not a proper circular-arc graph. It follows that the complement of an induced even cycle of length ≥ 6, or the complement of one of the graphs in Fig. 3, is a certificate to the fact that G is not a proper circular-arc graph, regardless of whether G is co-bipartite or not.

It is easy to see that the complement of each of the graphs of Fig. 3 is not a proper circular-arc graph. Each of these graphs has seven vertices, and any possible model for six of the vertices cannot be extended to accommodate the seventh arc.

By transforming a complement of an induced even cycle into an odd cycle in an incompatibility graph of G, we avoid using a complement of an induced even cycle as a certificate.

We handle the case where G is co-bipartite using a certifying algorithm of Hell and Huang [7] for proper interval bigraphs. The connection between the two graph families was also established by Hell and Huang [8].

Theorem 4 ([8]). *Let G be a co-bipartite graph. Then, G is a proper circular-arc graph if and only if $\overline{G}$ is a proper interval bigraph.*

We note that the graphs in Theorem 3 are exactly the graphs that the certifying algorithm for recognizing proper interval bigraphs [7] uses as certificates.

To summarize, the certificate for an answer of the algorithm is of one of three kinds. If G is a proper circular-arc graph, the certificate is a proper circular-arc model for it. If G is not a proper circular-arc graph then we use either Theorem 1 or Theorem 3 to provide a certificate which is either an odd length cycle in an incompatibility graph of G, or a complement of one of the graphs in Fig. 3 as an induced subgraph in G.

We begin the algorithm by deciding whether G is co-bipartite. If G is co-bipartite, then it is covered by two cliques. At least one of these cliques contains at least half of $V(G)$, so $m \geq \frac{n}{2}(\frac{n}{2} - 1)$. If this inequality does not hold then G is not co-bipartite. Otherwise, $m = \Theta(n^2)$, and we check if $\overline{G}$ is bipartite in $O(n^2) = O(m)$ time.

3.2. *The complement of G is not bipartite*

If G is not co-bipartite, then we use Theorem 2 and check whether $M^*(G)$ has the circular-ones property. We use the following reduction from testing the circular-ones property to testing the consecutive-ones property.

Theorem 5 (*[22]*). *Let M_1 be a $(0, 1)$-matrix. Fix a column j. Form the matrix M_2 by complementing those rows with 1 in column j of M_1. Then M_1 has the circular-ones property if and only if M_2 has the consecutive-ones property.*

Let v_0 be a vertex of minimum degree in G. To perform the reduction of Theorem 5 in linear time, we complement the rows of $M^*(G)$ which have one in the column of v_0. Since the degree of v_0 is $O(m/n)$, we complement $O(m/n)$ rows. It takes $O(n)$ time to complement a single row, so we perform the entire reduction in $O(m)$ time. We denote by M the new matrix that we obtain.

After the reduction we run the algorithm of McConnell [15] to test if M has the consecutive-ones property. If M has a consecutive-ones arrangement, we order the columns of $M^*(G)$ in the same way, to get a circular-ones arrangement for $M^*(G)$. We order the rows of $M^*(G)$ accordingly, since it is an adjacency matrix. Tucker [22] showed how to produce a proper circular-arc model of G from a circular-ones arrangement of $M^*(G)$. Tucker's algorithm can be implemented in $O(n + m)$ time.

If M does not have the consecutive-ones property, then the algorithm of [15] produces a certificate for this fact. This certificate is an odd cycle C of length at most $n + 2$ in the incompatibility graph $IC(M)$. Next, we show that all edges of C exist in $IC(G; v_0)$ so C is an odd cycle also in the incompatibility graph $IC(G; v_0)$.

Edges of C in $IC(M)$ have one of two forms. Edges of the form $(x, y)(y, x)$ always exist in $IC(G; v_0)$. Consider an edge $(x, y)(y, z)$ with a witness w, where w is a row in M such that the columns of x and z have 1 in this row, but the column of y has 0 in it. If $w \notin N[v_0]$ then the row of w in M is the same as in $M^*(G)$. So, $x, z \in N[w]$ while $y \notin N[w]$ and therefore $(x, y)(y, z)$ is an edge of $IC(G; v_0)$, with the vertex $w \notin N[v_0]$ as a witness. Otherwise, if $w \in N[v_0]$, then the row of w in M is the complement of the row of w in $M^*(G)$. So, $y \in N[w]$ while $x, z \notin N[w]$ and therefore $(x, y)(y, z)$ is an edge of $IC(G; v_0)$, with the vertex $w \in N[v_0]$ as a witness.

We provide the odd cycle C in $IC(G; v_0)$, together with v_0 as a certificate. To complete the certificate, we need to add a certificate for all edges and non-edges of G that are involved in it. For an edge $(x, y)(y, z)$ with a witness w in $IC(G; v_0)$, we need to provide a certificate for the edges or non-edges xw, yw, zw and wv_0 in G. The length of the cycle in $IC(G; v_0)$ is $O(n)$, and thus there are $O(n)$ edges or non-edges to certify.

3.3. The complement of G is bipartite

Recall that in this case, by Theorem 4, G is a proper circular-arc graph if and only if $\overline{G}$ is a proper interval bigraph. We apply the certifying algorithm of Hell and Huang [7] for proper interval bigraphs to $\overline{G}$. The graph G is covered by two cliques, one of these two cliques must cover at least $n/2$ of the vertices of G, therefore $m = \Theta(n^2)$. So we can produce $\overline{G}$ from G in $O(n^2) = O(m)$ time.

If $\overline{G}$ is an interval bigraph, we get an interval bigraph model for it, and we use an algorithm of Hell and Huang [8] to construct a proper circular-arc model of G, from this model of $\overline{G}$.

Otherwise, if $\overline{G}$ is not an interval bigraph, then we have one of graphs in Fig. 3, or an even induced cycle of length ≥ 6 in $\overline{G}$, as a certificate. For a graph of Fig. 3, we use its complement to certify that G is not a proper circular-arc graph.

If we have an induced even cycle of length ≥ 6 as a certificate that $\overline{G}$ is not a proper interval bigraph, we transform it into an odd cycle in an incompatibility graph of G. We do so for two reasons. First, a straightforward authentication of an even length cycle takes $O(n+m)$ time, while the authentication of an odd cycle in an incompatibility graph takes $O(n)$ time. Second, we reduce the number of cases that the authentication algorithm has to deal with, since it already has to verify an odd cycle in an incompatibility graph in the case where G is not co-bipartite. This makes the authentication algorithm simpler, as it should be.

Let $(x_0, x_1, \ldots, x_{2r-1})$ be an even induced cycle in $\overline{G}$ of length ≥ 6. For every $i = 0, \ldots, 2r - 1$, and for every $j \neq i \pm 1$, we have that $(x_i, x_j) \in E(G)$, where arithmetic on subscripts of the vertices is modulo $2r$. We find an odd cycle C in the incompatibility graph $IC(G; x_1)$.

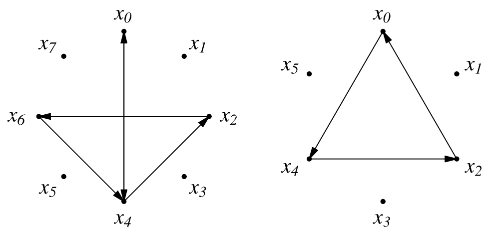
In the case where r is even (see Fig. 4(a)), $r \geq 4$, we start the cycle C in the incompatibility graph with (x_0, x_r). From the vertex (x_i, x_j) in C, we continue to (x_j, x_{i+2}). We can use x_{i+1} as a witness for the edge $(x_i, x_j)(x_j, x_{i+2})$, since if we start with (x_0, x_r), we always have $x_{i+1} \in N[x_1]$ and $x_i, x_{i+2} \notin N[x_{i+1}]$ while $x_j \in N[x_{i+1}]$. After r edges we get to (x_r, x_0), and we add the edge $(x_r, x_0)(x_0, x_r)$ to complete C as an odd cycle of length $r + 1$.

If r is odd (see Fig. 4(b)), $r \geq 3$, we start the cycle C in the incompatibility graph with (x_0, x_{r+1}). Again, from the vertex (x_i, x_j) in C, we continue to (x_j, x_{i+2}). As before, we can use x_{i+1} as a witness for the edge $(x_i, x_j)(x_j, x_{i+2})$, since if we start with x_0 and x_{r+1}, we always have $x_{i+1} \in N[x_1]$ and $x_i, x_{i+2} \notin N[x_{i+1}]$ while $x_j \in N[x_{i+1}]$. After r edges we get back to (x_0, x_{r+1}), and C is an odd cycle of length r.

Constructing the cycle C in the incompatibility graph $IC(G; x_1)$, together with certificate for all the edges of C takes $O(n + m)$ time.

3.4. The authentication algorithm

The certificate that the recognition algorithm provides is either a proper circular-arc model of G, an odd cycle in an incompatibility graph $IC(G; v_0)$, or a complement of one of the graphs of Fig. 3 as an induced subgraph of G. If we get a proper



(a) $r = 4$. (b) $r = 3$.

Fig. 4. Transforming the even cycle in $\overline{G}$ to an odd cycle in $IC(G; x_1)$. An arrow from x_i to x_j corresponds to the vertex (x_i, x_j) in the odd cycle in $IC(G; x_1)$.

circular-arc model of G, then G is a proper circular-arc graph. If we get one of the other certificates then by Theorem 1 or Theorem 3, G is not a proper circular-arc graph.

We authenticate in $O(m + n)$ time that a proper circular-arc model is a circular-arc model of G as described by McConnell [14]. We initialize an empty list of intersections. We start at an arbitrary point on the circle, and produce a (doubly-linked) list L of all arcs that contain this point by going to the right and around the circle. Every arc whose right endpoint is traversed before its left endpoint is inserted to L. Then we go again around the circle, in the same direction, maintaining L to be the list of arcs that cover the endpoint which we traverse next. Let e be the next endpoint which we traverse. If e is a left endpoint we add the corresponding arc to L, and if e is a right endpoint we remove the corresponding arc from L. In both cases we add intersections between the arc of e and every arc of L to the list of intersections. We stop accumulating intersections either when we have more than $4m$ of them, or when we return to our starting point on the circle. If we found more than $4m$ intersections then the model cannot be of G, because each pair of arcs has a total of four endpoints, and each intersection is found at most once for each endpoint. Otherwise, we compare the list of intersections that we found to the list of adjacencies in G, by radix sorting the list we found and eliminating duplicates.

After we verified that the model is indeed a model of G, we verify that it is a proper circular-arc model. We check for every pair of adjacent vertices that any of their corresponding arcs does not contain the other. We do so by checking the order of the endpoints of the arcs. We conclude that in the case where G is a proper circular-arc graph the size of the certificate, which is a proper circular-arc model of G, is $O(n)$, and the time to authenticate it is $O(n + m)$.

To authenticate an odd cycle in an incompatibility graph $IC(G; v_0)$, we first verify that it has an odd length not larger than $n+2$. Then, we verify that the certificate is indeed a cycle. We also verify that each edge of the cycle belongs to $IC(G; v_0)$, by checking that every edge is either of the form $(x, y)(y, x)$ or has a valid witness. The size of the cycle is $O(n)$ and validating it takes $O(n)$ time.

If the certificate is a complement of one of the graphs of Fig. 3 as an induced subgraph in G, we verify that every edge exists in the certificate if and only if it exists in G. The size of each of these graphs is $O(1)$, and hence the authentication time is also $O(1)$.

When the algorithm found that G is not a proper circular-arc graph, both possible certificates can be authenticated in $O(n)$ time, therefore the certificate in this case is a strong certificate.

4. Certifying algorithm for recognizing unit circular-arc graphs

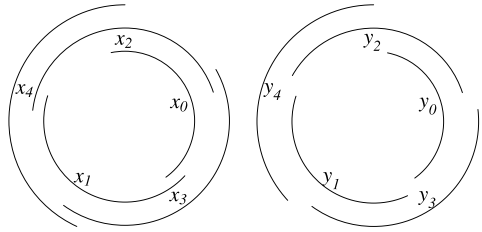
4.1. A characterization of unit circular-arc graphs

In this section we present the structure theorem of Tucker [23] for unit circular-arc graphs. Note that every proper circular-arc graph has a proper circular-arc model in which no pair of arcs covers the circle [23,6], we discuss this further in the next section, and assume that all models we work with here satisfy this property. Let G be a proper circular-arc graph with a proper circular-arc model ϱ.

Let $L = (x_0, \ldots, x_{p-1})$ be a list of vertices of G. Assume that we traverse the cyclic list of endpoints of ϱ, starting immediately after $\ell(x_0)$, going right to $r(x_0)$ and continuing from $r(x_i)$ to $r(x_{i+1})$ until we get to $r(x_{p-1})$. We call the list of endpoints that we encounter in this traversal the *walk* of L.

Let $C = (x_0, \ldots, x_{p-1})$ be a simple cycle in G, such that for $i = 1, \ldots, p-1$, the arc x_i overlaps the right side of x_{i-1}, and the arc x_0 overlaps the right side of x_{p-1}, in ϱ. We call such a cycle C, a *bounding cycle*. The number of times that C goes around the circle is the number of times that the walk of C hits $\ell(x_0)$, we denote this number by $\textsc{turns}(C)$. The *ratio* of C, denoted by $\textsc{ratio}(C)$, is $|C|/\textsc{turns}(C)$. (See Fig. 5(a)). We call C a *minimum bounding cycle* if there is no other bounding cycle C' with $\textsc{ratio}(C') < \textsc{ratio}(C)$. We denote by C^m an arbitrary minimum bounding cycle. If the union of the arcs in ϱ does not cover the circle, then there are no bounding cycles. In this case, we define $C^m = \emptyset$ and $\textsc{ratio}(\emptyset) = \infty$.

Let $I = (x_0, \ldots, x_{p-1})$ be a simple co-cycle in G. We call I a *bounding co-cycle*. To compute $\textsc{turns}(I)$, the number of times that I goes around the circle, we add 1 to the number of times that the walk of I hit $\ell(x_0)$, to count also the last partial turn. The *ratio* of I, denoted by $\textsc{ratio}(I)$, is $|I|/\textsc{turns}(I)$. (See Fig. 5(b)). We call I a *maximum bounding co-cycle* if there is no other bounding co-cycle I' with $\textsc{ratio}(I') > \textsc{ratio}(I)$. We denote by I^M an arbitrary maximum bounding co-cycle.



(a) Bounding cycle. (b) Bounding co-cycle.

Fig. 5. Bounding cycle and bounding co-cycle, both with ratio 5/2.

The circumference of a unit circular-arc model of closed arcs with a bounding cycle C can be at most $\text{RATIO}(C)$. On the other hand, the circumference of a unit circular-arc model with a bounding co-cycle I must be strictly greater than $\text{RATIO}(I)$. So for any unit circular-arc model $\text{RATIO}(I^M) < \text{RATIO}(C^m)$. The following theorem shows that this condition is also sufficient. Furthermore, the bounds do not depend on the specific model.

Theorem 6 (*[23]*). *Let G be a proper circular-arc graph. If there exists a proper circular-arc model of G with no pair of arcs that covers the circle, such that* $\text{RATIO}(I^M) < \text{RATIO}(C^m)$*, then G is a unit circular-arc graph. If G is a unit circular-arc graph, then in every proper circular-arc model of G with no pair of arcs that covers the circle,* $\text{RATIO}(I^M) < \text{RATIO}(C^m)$.

The following lemma describes the structure of a minimum bounding cycle and a maximum bounding co-cycle if G is a proper circular-arc graph which is not a unit circular-arc graph.

Lemma 7 (*[23]*). *Let G be a proper circular-arc graph. Let* C^m *be a minimum bounding cycle and let* I^M *be a maximum bounding co-cycle in a proper circular-arc model of G with no pair of arcs that covers the circle. If* $\text{RATIO}(I^M) \geq \text{RATIO}(C^m)$ *then* $\text{RATIO}(I^M) = \text{RATIO}(C^m)$, $|I^M| = |C^m|$*, and* C^m *and* I^M *do not share a common vertex.*

4.2. Pairs of arcs that cover the circle

In the previous section we presented a characterization of unit circular-arc graphs that is defined for proper circular-arc models with no pair of arcs that covers the circle. The following theorem shows that this requirement is valid for any proper circular-arc graph.

Theorem 8 (*[23,6]*). *If G is a proper circular-arc graph, then G has a proper circular-arc model in which no pair of arcs covers the circle.*

Let x and y be a pair of arcs that covers the circle in ϱ. Any arc in ϱ that is disjoint from x must be contained in y, but since ϱ is a proper circular-arc model, there is no such arc. So every arc in a pair that covers the circle represents a universal vertex in G.

Let u be a universal vertex in G. In any proper circular-arc model of G, every arc must either cover $\ell(u)$ or $r(u)$. The vertices of the arcs that cover each of the endpoints of u define a clique in G. Therefore, G is co-bipartite.

In the algorithm of Section 3, if we receive as an input a co-bipartite graph G then we use the algorithms of [8] to construct a proper circular-arc model for it. The algorithm of [8] always builds a proper circular-arc model without a pair of arcs that covers the circle. Therefore, our algorithm never constructs a model with a pair of arcs that covers the circle.

If we get as an input a proper circular-arc model, in which there might be a pair of arcs which covers the circle, we can use the observation that every arc in a pair that covers the circle is an arc of a universal vertex, to give a simple $O(n)$ time algorithm to eliminate pairs of arcs that cover the circle in the given model. We go around the circle twice, in a left to right direction. In the first round, we count the number of arcs that cover the starting point, this is the number of arcs for which we traversed the right endpoint before the left endpoint. In the second round, we find the number of arcs that cover each endpoint. We maintain a counter of the number of arcs that contain our current position. When we encounter a left endpoint we increment the counter. When we encounter a right endpoint we decrement the counter. Since any arc which does not correspond to a universal vertex cannot cover the circle with another arc, the sum of the values of the counter at the endpoints of an arc x is greater or equal to $n - 1$, if and only if x is a universal vertex. After we identified all the arcs of universal vertices, we pick arbitrarily one of these arcs, and put all other arcs of universal vertices in the same place on the circle, with a slight shift.

4.2.1. Co-bipartite unit circular-arc graphs

We show that the class of co-bipartite proper circular-arc graphs is equivalent to the class of co-bipartite unit circular-arc graphs. As mentioned earlier, the class of the complements of co-bipartite proper circular-arc graphs is known to be equivalent to many other well-known graph classes including bipartite permutation graphs, bipartite AT-free graphs and bipartite trapezoid graphs.

Theorem 9. *A graph G is a co-bipartite proper circular-arc graph if and only if G is a co-bipartite unit circular-arc graph.*

Proof. Assume that G is a co-bipartite proper circular-arc graph. For any proper circular-arc model of G, $\text{RATIO}(I^M) \le 2$. This is because we can traverse at most two disjoint arcs in any turn around the circle. For any proper circular-arc model of G without pair of arcs that covers the circle, $\text{RATIO}(C^m) > 2$, and by Theorem 8, such a model exists. So from Theorem 6 it follows that G is a co-bipartite unit circular-arc graph.

Assume that G is a co-bipartite unit circular-arc graph. Since every unit circular-arc graph is a proper circular-arc graph it follows that G is a co-bipartite proper circular-arc graph. □

Since a proper circular-arc graph with a universal vertex is co-bipartite, it follows from Theorem 9 that every proper circular-arc graph with a universal vertex is also a unit circular-arc graph.

4.3. Easy-to-find minimum bounding cycles and maximum bounding co-cycles

Let G be a proper circular-arc graph with a proper circular-arc model ϱ. By the previous section we assume that no pair of arcs covers the circle in ϱ.

In this section we show that every proper circular-arc model which satisfies some assumptions has a minimum bounding cycle and a maximum bounding co-cycle of a specific structure. This makes the task of finding such a minimum bounding cycle and a maximum bounding co-cycle easier.

First, we show that no matter where we start a cycle the ratio is the same.

Lemma 10. *Let $C = (x_0, \dots, x_{p-1})$ be a minimum bounding cycle. Let $C' = (x_1, \dots, x_{p-1}, x_0)$ be the bounding cycle which is derived from C by moving x_0 to the end. Then C' is also a minimum bounding cycle.*

Proof. Let $C'' = (x_1, \dots, x_{p-1})$ the list of vertices we obtain by removing the first vertex from C.

Assume that x_1 overlaps the right side of x_{p-1}. In this case, C'' is a bounding cycle. The number of times that $\ell(x_1)$ appears in the walk of C is $\text{TURNS}(C) + 1$, since every instance of $\ell(x_0)$ in this walk is followed by an instance of $\ell(x_1)$, and the first instance of $\ell(x_1)$ is not preceded by $\ell(x_0)$. The walk of C'' is the suffix of the walk of C which starts after the first instance of $\ell(x_1)$, therefore $\ell(x_1)$ appears $\text{TURNS}(C)$ times in the walk of C''. So $\text{TURNS}(C'') = \text{TURNS}(C)$ and $|C''| = |C| - 1$, and therefore $\text{RATIO}(C'') < \text{RATIO}(C)$, contradicting the fact that C is a minimum bounding cycle. So, x_1 does not overlap the right side of x_{p-1} and $\ell(x_1)$ appears $\text{TURNS}(C) - 1$ times in the walk of C'', since the last instance of $\ell(x_0)$ is not followed by $\ell(x_1)$ in the walk.

The walk of C' is obtained from the walk of C'' by appending to it the endpoints after $r(x_{p-1})$ and to right up to $r(x_0)$. The endpoint $\ell(x_1)$ is among these endpoints and so $\ell(x_1)$ appears $\text{TURNS}(C)$ times in the walk of C'. It follows that $\text{TURNS}(C') = \text{TURNS}(C)$, $|C'| = |C|$, and $\text{RATIO}(C') = \text{RATIO}(C)$. Since C is a minimum bounding cycle, so is C'. □

Let C be a cycle and let $k \ge 1$. We denote by kC the list of vertices that is obtained by concatenating the list of C to itself k times. We define $\text{TURNS}(kC)$ as we defined $\text{TURNS}(C)$ to be the number of times which the walk of kC hits $\ell(x_0)$ where x_0 is the first vertex of C. We also define the ratio of kC to be $|kC|/\text{TURNS}(kC)$.

Lemma 11. *Let $C = (x_0, \dots, x_{p-1})$ be a bounding cycle and let $k \ge 1$. Then, $\text{RATIO}(kC) = \text{RATIO}(C)$.*

Proof. For $k = 1$, the cycles C and kC are identical and so $\text{RATIO}(kC) = \text{RATIO}(C)$.

Assume, by induction, that $\text{RATIO}((k-1)C) = \text{RATIO}(C)$. Since $|(k-1)C| = (k-1)|C|$ then $\text{TURNS}((k-1)C) = (k-1)\text{TURNS}(C)$. The walk of kC is obtained from the walk of $(k-1)C$ by appending to it the suffix of the walk of C which starts after the first instance of $r(x_{p-1})$, we denote this suffix by S. Since the walk of C starts after $\ell(x_0)$, all the occurrences of $\ell(x_0)$ in the walk of C are also in S. Therefore, S hits $\text{TURNS}(C)$ times the endpoint $\ell(x_0)$, and so the walk of kC hits this endpoint $\text{TURNS}((k-1)C) + \text{TURNS}(C)$ times. We got that $\text{RATIO}(kC) = \frac{|kC|}{\text{TURNS}(kC)} = \frac{|kC|}{\text{TURNS}((k-1)C)+\text{TURNS}(C)} = \frac{k|C|}{k\text{TURNS}(C)} = \frac{|C|}{\text{TURNS}(C)} = \text{RATIO}(C)$. □

For each vertex $v \in V(G)$, we define $\text{NEXT}(v)$ to be the vertex u, where $\ell(u)$ is the rightmost left endpoint covered by the arc v. If the circle is not covered by the union of the arcs then $\text{NEXT}(v)$ is not always defined. But, in this case ϱ is a *proper interval model* and therefore G is *a proper interval graph*. Since a graph is a proper interval graph if and only if it is a unit interval graph [19] it follows that G is also a *unit interval graph* and therefore a unit circular-arc graph.

Intuitively, the arc of $\text{NEXT}(v)$ is the arc which extends to the right as far as possible from the arc v. So if each vertex x_i in a bounding cycle C is followed by $\text{NEXT}(x_i)$, then $\text{TURNS}(C)$ is the maximum possible for a bounding cycle with length $|C|$ which starts at the same vertex, and therefore $\text{RATIO}(C)$ is minimum for such bounding cycle. The following lemma shows that there is always a minimum bounding cycle of this form.

Lemma 12. *Let G be a proper circular-arc graph and ϱ be a proper circular-arc model of G without a pair of arcs that cover the circle, and such that the union of the arcs of ϱ covers the circle. There is a minimum bounding cycle $C = (x_0, \dots, x_{p-1})$ such that for $i = 1, \dots, p-1$, $x_i = \text{NEXT}(x_{i-1})$ and $x_0 = \text{NEXT}(x_{p-1})$.*

Proof. Let $C_0 = (c_0, \ldots, c_{r-1})$ be a shortest minimum bounding cycle (the ratio of C_0 is $\textsc{ratio}(C^m)$). Since there is no pair of arcs which covers the circle then $|C_0| > 2$.

Let $F = (f_0, f_1, f_2, \ldots)$ be the list of vertices such that $f_0 = c_0$ and $f_i = \textsc{next}(f_{i-1})$ for $i \geq 1$. Since the size of G is finite, there are i, j such that $i \neq j$ and $f_i = f_j$. Let f_s be the first vertex in F for which there exists a vertex f_j such that $s < j$ and $f_s = f_j$ and let f_t be the first vertex in F such that $s < t$ and $f_s = f_t$. We show that $F' = (f_s, \ldots, f_{t-1})$ is a minimal bounding cycle. The fact that F' is a bounding cycle follows from its definition.

Replace c_1 in C_0 with $f_1 = \textsc{next}(c_0)$, and call the resulting list of vertices C'_0. We show that C'_0 is a minimum bounding cycle.

The arc f_1 overlaps the right side of c_0. By the way we defined $\textsc{next}(c_0)$, this arc also overlaps the right side of c_1. Since c_2 also overlaps the right side of c_1, it intersects f_1, so C_1 is a cycle.

We show that c_2 must overlap the right side of f_1. If c_2 overlaps the right side of c_0 then we can get a bounding cycle S by removing c_1 from C_0 such that the walk of S is identical the walk of C_0. Since $|S| < |C_0|$ we get that $\textsc{ratio}(S) < \textsc{ratio}(C_0)$ which contradicts the fact that C_0 is a minimum bounding cycle. So c_2 does not overlap the right side of c_0, and therefore it must overlap the right side of f_1.

Assume that there is a vertex c_j in C'_0 such that $c_j = f_1$, in this case C'_0 is not a simple cycle. Let us split C'_0 into two parts—$C_a = (c_2, \ldots, c_j)$ and $C_b = (c_0, f_1, c_{j+1}, \ldots, c_{r-1})$ (if $j = r - 1$, then the last vertex of C_b would be f_1). Both C_a and C_b are simple cycles, and since we showed that c_2 overlaps the right side of $f_1 = c_j$, both are bounding cycles.

The cycle C_a is shorter than the cycle C_0, and since C_0 is a shortest minimum bounding cycle, we know that $\textsc{ratio}(C_a) > \textsc{ratio}(C^m)$. The same is true for C_b, that is $\textsc{ratio}(C_b) > \textsc{ratio}(C^m)$.

The walk of C_a goes around the circle $\textsc{turns}(C_a)$ times starting after $\ell(c_2)$ and ending at $r(c_j)$. Let W_a be the suffix of the walk of C_a starting after the first instance of $r(f_1) = r(c_j)$ in this walk. Every endpoint, including $\ell(c_0)$, appears $\textsc{turns}(C_a)$ times in W_a. The walk of C_b hits the endpoint $\ell(c_0)$ $\textsc{turns}(C_b)$ times. Let W_b be the prefix of the walk of C_b starting after $\ell(c_0)$ and ending at the first instance of $r(f_1)$, and let W'_b be the suffix of the walk of C_b which starts after the first instance of $r(f_1)$. The walk of C_0 is exactly the concatenation of W_b, W_a and W'_b, therefore $\ell(c_0)$ appear in this walk $\textsc{turns}(C_b) + \textsc{turns}(C_a)$ times and so $\textsc{turns}(C_0) = \textsc{turns}(C_b) + \textsc{turns}(C_a)$. Also, $|C_0| = |C_a| + |C_b|$, and since $\textsc{ratio}(C_b) > \textsc{ratio}(C^m)$ and $\textsc{ratio}(C_a) > \textsc{ratio}(C^m)$, we get that $\textsc{ratio}(C_0) > \textsc{ratio}(C^m)$. This is a contradiction to the fact that C_0 is a minimum bounding cycle. Therefore C'_0 is a simple cycle.

Since C'_0 is a simple cycle, c_2 overlaps the right side of f_1, and f_1 overlaps the right side of c_0, it follows that C'_0 is a bounding cycle. Since we replace one arc by another, $|C'_0| = |C_0|$, and since the walk of C'_0 is identical to the walk of C_0, $\textsc{turns}(C'_0) = \textsc{turns}(C_0)$. So, $\textsc{ratio}(C'_0) = \textsc{ratio}(C_0) = \textsc{ratio}(C^m)$ and C'_0 is a minimum bounding cycle of shortest length.

Let C_1 be the bounding cycle $(f_1, c_2, \ldots, c_{r-1}, c_0)$ which is obtained by from C'_0 by moving the first vertex to the end. By Lemma 10, C_1 is also a shortest minimum bounding cycle. We repeat this argument starting with C_1 and get a shortest bounding cycle, C_2, that starts with f_2. After repeating this argument s times we get a shortest minimum bounding cycle C_s which starts with f_s.

The lists of vertices $|F'|C_s$ and $|C_s|F'$ both start with f_s and have the same length. In the list $|C_s|F'$, the successor of each vertex f is the vertex $\textsc{next}(f)$ whose arc covers the highest number of endpoints after $r(f)$, therefore the walk of $|F'|C_s$ is a prefix of the walk of $|C_s|F'$. So, $\textsc{turns}(|F'|C_s) \leq \textsc{turns}(|C_s|F)$, and since the length of the two lists is the same we get that $\textsc{ratio}(|F'|C_s) \geq \textsc{ratio}(|C_s|F')$. By Lemma 11, we get that $\textsc{ratio}(F') \leq \textsc{ratio}(C_s)$, and since C_s is a minimum bounding cycle, so is F'. $\square$

For each vertex $v \in V(G)$, we define $\textsc{next}'(v)$ to be the vertex u, where $\ell(u)$ is the leftmost left endpoint not covered by the arc v. If for some vertex v, the arc of $\textsc{next}'(v)$ intersects v, then v is a universal vertex. In this case, by the discussion in Section 4.2.1, G is a unit circular-arc graph.

Intuitively, the arc of $\textsc{next}'(v)$ is the arc which goes least to the right from the arc v. So if each vertex x_i in a bounding co-cycle I is followed by $\textsc{next}'(x_i)$, then $\textsc{turns}(I)$ is the minimum possible for a bounding co-cycle with length $|I|$ which starts with the same vertex, and therefore $\textsc{ratio}(I)$ is maximum for such bounding co-cycle. The following lemma is similar to Lemma 12 and shows that there always exists a co-cycle of this form. The proof of this lemma is symmetric to the proof of Lemma 12 by replacing $\textsc{next}(\cdot)$ with $\textsc{next}'(\cdot)$ and minimum with maximum (we also need lemmata analogous to Lemma 10 and Lemma 11 for co-cycles).

Lemma 13. *Let G be a proper circular-arc graph without a universal vertex and let ϱ be a proper circular-arc model of G without pair of arcs that cover the circle. There is a maximum bounding co-cycle $I = (x_0, \ldots, x_{p-1})$ such that for $i = 1, \ldots, p-1$, $x_i = \textsc{next}'(x_{i-1})$ and $x_0 = \textsc{next}'(x_{p-1})$.*

4.4. Recognition algorithm for unit circular-arc graphs

Recall that every unit circular-arc graph is a proper circular-arc graph. So the algorithm which we describe takes as input either a graph G represented by adjacency lists of its edges or a graph G represented by a proper circular-arc model. In the first case, when we do not get a proper circular-arc model, we start by testing whether G is a proper circular-arc graph using the algorithm of Section 3. If G is not a proper circular-arc graph then it is also not a unit circular-arc graph, and the algorithm of Section 3 certifies that. Otherwise, if G is a proper circular-arc graph then we have a proper circular-arc model of it which we denote by ϱ. As observed in Section 4.2, there is no pair of arcs in ϱ that covers the circle. Note that obtaining the proper circular-arc model of G is the only step of the recognition algorithm which takes $O(n+m)$ time. If the input graph G is given by a proper circular-arc model then we begin by eliminating pairs of arcs which cover the circle, as described in Section 4.2. So when we finish this first stage we have a proper circular-arc model of G in which no pair of arcs covers the circle, and the rest of the algorithm runs in $O(n)$ time.

A high level description of the rest of the algorithm is as follows. We find a minimum bounding cycle and a maximum bounding co-cycle in ϱ using the characterizations of Section 4.3. We compare the ratios of the bounding cycle and the bounding co-cycle, and use Theorem 6 to decide whether G is a unit circular-arc graph. If the ratio of the minimum bounding cycle is strictly greater than the ratio of the maximum bounding co-cycle then G is a unit circular-arc graph. Otherwise, we provide the minimum bounding cycle and the maximum bounding co-cycle, as a certificate for the answer that G is not a unit circular-arc graph.

To find a minimum bounding cycle and a maximum bounding co-cycle, we find for each vertex $v \in V(G)$ the vertices $\textsc{next}(v)$ and $\textsc{next}'(v)$. We do so in $O(n)$ time by going around the circle from left to right, starting at some left endpoint, and maintaining $\ell(u)$, the last left endpoint which we encountered. When we encounter a right endpoint $r(v)$, then $\textsc{next}(v) = u$. We can find $\textsc{next}'(v)$ for each v in the same way, by going around the circle from right to left.

If for some vertex v, we found $\textsc{next}(v) = v$, then the union of the arcs does not cover the circle, and G is a unit circular-arc graph as discussed in Section 4.3. On the other hand, if we find $\textsc{next}(v)$ for every vertex v, then the union of the arcs covers the circle. If for some vertex v, the arc $\textsc{next}'(v)$ intersects v then v is a universal vertex and G is a unit circular-arc graph, as discussed in Section 4.2.1. On the other hand, if for every vertex v the arc of $\textsc{next}'(v)$ does not intersect the arc of v, then G does not have a universal vertex.

So at this point we either know that the graph is a unit circular-arc graph, or we have computed $\textsc{next}(v)$ and $\textsc{next}'(v)$ for every vertex v and we continue to find a minimum bounding cycle and a maximum bounding co-cycle as follows.

Let D be the directed graph whose set of vertices is $V(G)$ and its set of directed edges are $\{(v, u) \mid u = \textsc{next}(v)\}$. The outdegree of each vertex in D is 1. Therefore, we can detect all directed cycles in D in $O(n)$ time.

Each directed cycle in D represents a bounding cycle in ϱ. By Lemma 12, at least one of these bounding cycles is minimal. By Lemma 10 it does not matter which vertex we choose to be the first in a given minimal bounding cycle. So, we start each directed cycle in D at an arbitrary vertex.

Let $C = (x_0, \dots, x_{p-1})$ be one of the directed cycles in D and therefore also a bounding cycle in ϱ. The ratio of C is equal to $\textsc{turns}(C)/p$ where $\textsc{turns}(C)$ is the number of times that the walk of C hits $\ell(x_0)$. We compute $\textsc{turns}(C)$ by counting the number of vertices x_i, $0 < i \le p-1$, in C, for which $\ell(x_0)$ is between $r(x_{i-1})$ and $r(x_i)$ when going from left to right around the circle. This computation takes $O(|C|)$ time. We find the ratio of every bounding cycle this way, and a cycle with the lowest ratio is a minimal bounding cycle. Since each vertex belongs to at most one directed cycle in D, then the sum of the lengths of all cycles in D is at most n, and it takes $O(n)$ time to identify the minimum bounding cycle among the directed cycles of D.

Similarly, let D' be the directed graph whose set of vertices is $V(G)$ and its set of directed edges are $\{(v, u) \mid u = \textsc{next}'(v)\}$. The outdegree of each vertex in D' is 1 so we can compute all cycles in D' in $O(n)$ time. By Lemma 13 one of these cycles represents a maximum bounding co-cycle in ϱ.

Let $I = (x_0, \dots, x_{p-1})$ be a directed cycle in D' and therefore also a bounding co-cycle in ϱ. The ratio of I equals to $\textsc{turns}(I)/p$ where $\textsc{turns}(I)$ is the number of times that the walk of I hits $\ell(x_0)$ plus one. We compute $\textsc{turns}(I)$ by counting the number of vertices x_i, $0 < i \le p-1$, in I, for which $\ell(x_0)$ is between $r(x_{i-1})$ and $r(x_i)$ when going from left to right around the circle, and adding one to the total count. This takes $O(|I|)$ time. We find the ratio of every bounding co-cycle, and a co-cycle with the highest ratio is a maximum bounding co-cycle. Since each vertex belongs to at most one directed cycle in D', then the sum of the lengths of all cycles in D' is at most n, and therefore it takes $O(n)$ time to compute a maximum bounding co-cycle from the directed cycles of D'.

Let C be the minimum bounding cycle that we found and let I be the maximum bounding co-cycle that we found. From Theorem 6 follows that G is a unit circular-arc graph if and only if $\textsc{ratio}(I) < \textsc{ratio}(C)$.

4.5. Certificates for unit circular-arc graphs

If G is a unit circular-arc graph then a unit circular-arc model certifies this fact. If G is not a proper circular-arc graph, then we use the certificates from Section 3. The last case to handle is when G is a proper circular-arc graph but not a unit circular-arc graph.

Our recognition algorithm finds a minimum bounding cycle C and a maximum bounding co-cycle I in a proper circular-arc model ϱ of G. If G is a proper circular-arc graph, but not a unit circular-arc graph, then the model ϱ, together with C and I certify that G is not a unit circular-arc graph. If ϱ is part of the input, then it is easy to verify in $O(n)$ time that the bounding cycle C and the bounding co-cycle I are indeed bounding cycle and bounding co-cycle with the same ratio in ϱ (note that if G is not a unit circular-arc graph then our algorithm does not change ϱ). However, if the input is given by adjacency lists of its edges and our algorithm produced ϱ, then it takes $O(n+m)$ time to verify that ϱ is indeed a proper circular-arc model of G.

Fortunately, we can provide a strong certificate which can prove in $O(n)$ time that G is not a unit circular-arc graph when it is represented by adjacency lists. While the description of the certificate is involved, the authentication algorithm

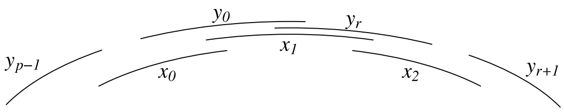
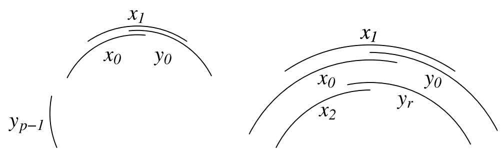


Fig. 6. Arcs certifying that x_2 overlaps the right side of x_1.



(a) The union of x_0 and y_0 must contain x_1.

(b) The union of x_2 and y_r also contains x_1.

Fig. 7. x_1 overlaps the right side of both x_0 and x_2.

itself is simple. All it does is authenticating $O(n)$ edges and non-edges of G. As mentioned in Section 2, construction of such a certificate takes $O(n+m)$ time. For the strong certificate we use the following Theorem. For the rest of this section, arithmetic on subscripts of the vertices of C and I is modulo p.

Theorem 14. *Let G be a graph. Let $C = (x_0, \ldots x_{p-1})$ be a simple cycle of vertices in G and let $I = (y_0, \ldots y_{p-1})$ be a simple co-cycle of vertices in G. If there is an $0 \le r < p$, such that for every $0 \le i < p$, x_iy_i, $x_{i+1}y_i$, $x_{i+1}y_{i+r}$, $x_{i+2}y_{i+r}$ are all edges of G and x_iy_{i+r}, $x_{i+1}y_{i-1}$, $x_{i+1}y_{y+r+1}$, $x_{i+2}y_i$ are all non-edges of G, then G is not a unit circular-arc graph.*

An illustration of the arcs of the certificate, for $i = 0$, is given in Fig. 6.

Proof. Assume that G is a unit circular-arc graph. Then, the graph G has a proper circular-arc model ϱ, since it is also a proper circular-arc graph. We may assume that in ϱ the arc x_1 overlaps the right side of the arc x_0. Also, by Theorem 8, we assume that ϱ has no pair of arcs which covers the circle. We show that in this case C and I must be a bounding cycle and bounding co-cycle, respectively, with $\text{RATIO}(C) = \text{RATIO}(I)$. Then, Theorem 6 implies that G is not a unit circular-arc graph, in contradiction the assumption.

Every simple co-cycle of G is a bounding co-cycle, so I is a bounding co-cycle. To show that C is a bounding cycle we have to show that for every $x_i \in C$, the arc x_{i+1} overlaps the right side of the arc x_i in the model ϱ.

For $i = 0$, the arc $x_{i+1} = x_1$ overlaps the right side of $x_i = x_0$, by the way definition of ϱ. To complete the proof that C is a bounding cycle, we show that if the arc x_{i+1} overlaps the right side of the arc x_i in ϱ, then x_{i+2} overlaps the right side of x_{i+1} in ϱ. We show this for $i = 0$, that is we show that x_2 overlaps the right side of x_1 in ϱ. The same proof holds for any $0 \le i < p$.

Assume for a contradiction that the arc x_1 overlaps the right side of both x_0 and x_2. In this case, the arc y_0 overlaps the right side of x_1, since y_0 and x_2 are not adjacent. The arc y_r also overlaps the right side of x_1, since y_r and x_0 are not adjacent. The arcs x_0 and y_0 overlap each other, since the vertices x_0 and y_0 are adjacent. The arcs x_0 and y_0 cover different endpoints of x_1, so the union of these arcs either contains x_1 or covers the part of the circle that is not covered by x_1. The arcs x_1 and y_{p-1} are disjoint, so if the union of the arcs x_0 and y_0 cover the part of the circle that is not covered by x_1, then this union contains y_{p-1}. The arc x_0 cannot contain the arc y_{p-1}, in addition y_0 and y_{p-1} are not adjacent, since I is a co-cycle. Therefore, the union of the arcs x_0 and y_0 cannot contain y_{p-1}. So, the union of the arcs x_0 and y_0 contains the arc x_1, and hence the common part of the arcs is covered by x_1 (see Fig. 7(a)). Similarly, since the vertices x_2 and y_r are adjacent we get that the union of the arcs x_2 and y_r must contain the arc x_1, since it cannot contain y_{r+1}. Hence, the common part of the arcs is covered by x_1 (see Fig. 7(b)). Since x_0 and y_0 are not adjacent to y_r and x_2 respectively, we get a contradiction. Therefore, the arc x_2 overlaps the right side of x_1 and we conclude that C is a bounding cycle.

To complete the proof, we have to show that $\text{RATIO}(C) = \text{RATIO}(I)$. Since $|C| = |I|$, it is enough to verify that $\text{TURNS}(C) = \text{TURNS}(I)$.

For every i, the arc y_{i-1} overlaps the arc x_i, and is disjoint from x_{i+1}. Since x_{i+1} overlaps the right side of x_i, the arc y_{i-1} overlaps the left side of x_i. The arc y_i overlaps x_i and x_{i+1} but it is disjoint from y_{i-1}, so y_i overlaps the right side of x_i and the left side of x_{i+1}. Therefore, when we traverse the circle's cyclic list of endpoints, going right from $r(x_i)$ to $r(x_{i+1})$, as we do in the walk of C, we encounter $r(y_i)$ between these two endpoints.

The walk of C starts after $\ell(x_0)$ and ends at $r(x_{p-1})$. The walk of I starts after $\ell(y_0)$ and ends at $r(y_{p-1})$. Let S be the suffix of the walk of C that stars after the first instance of $\ell(y_0)$. We know that the left endpoints of x_i's and y_i's alternate in S, so S is a prefix of the walk of I.

Let t_x denote the number of occurrences of $\ell(x_0)$ in S, and let t_y denote the number of occurrences of $\ell(y_0)$ in S. We have $t_x = \text{TURNS}(C)$, since the part of the walk of C which is not in S is the prefix of the walk of C which ends at the first occurrence of $\ell(y_0)$, and this prefix does not contain occurrences of $\ell(x_0)$. We also have $t_y = \text{TURNS}(I) - 1$, since the only member of I

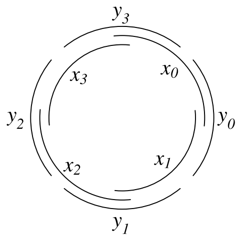


Fig. 8. A bounding cycle and a bounding co-cycle with the same ratio 4/1 in the same model.

in the part of the walk of I which is not in S is y_{p-1}, which is not adjacent to y_0. The endpoints $\ell(x_0)$ and $\ell(y_0)$ alternate in S. Moreover, the first occurrence of $\ell(x_0)$ in S is before the first occurrence of $\ell(y_0)$ in S, since S starts after $\ell(y_0)$. Also, the last occurrence of $\ell(x_0)$ in S is after the last occurrence of $\ell(y_0)$ in S, since S ends and $r(x_{p-1})$ and x_{p-1} overlap the left side of x_0 and is disjoint from y_0. Therefore $t_x = t_y + 1$. We got that $\text{TURNS}(C) = t_x = t_y + 1 = \text{TURNS}(I)$, as required. $\square$

4.6. Finding the certificate

If G is a unit circular-arc graph, we obtain a unit circular-arc model for it using the algorithm of [13]. Note, that the first two steps of [13] are to find a proper circular-arc model of G and to eliminate pairs of arcs that cover the circle from this model, so we do not need to run them again since we already have an appropriate proper circular-arc model. If G is not a proper circular-arc graph, then the certificate is as in Section 3. The size of these certificates is $O(n)$.

We now show that the recognition algorithm can provide the certificate required by Theorem 14 for the case where G is a proper circular-arc graph but not a unit circular-arc graph. For constructing the certificate, the algorithm uses the cycle C and the co-cycle I that it has found.

To certify that C is a simple cycle, we provide the $|C|$ edges of the form $x_i x_{i+1}$. Similarly, to certify that I is a simple co-cycle, we provide the $|I|$ non-edges of the form $y_i y_{i+1}$. By Lemma 7, we know that $|I| = |C|$, so this part of the certificate is of size $2|C|$.

In order to complete the certificate, we present few observations about the arcs of the vertices of C and I in ϱ. An example of a bounding cycle and a bounding co-cycle with the same ratio in the same model is illustrated in Fig. 8.

Observation 15. *For every arc x_i of a vertex in C, there is an arc y_j of a vertex in I, such that x_i does not overlap y_j.*

Proof. Assume that x_i overlaps all the arcs of the vertices of I. Then, I can be partitioned into two cliques, one consisting of the vertices whose arcs cover $\ell(x_i)$ and the other consisting of the vertices whose arcs cover $r(x_i)$. So every time the walk of I goes around the circle, it traverses at most two arcs of I, therefore $\text{RATIO}(I) = 2$. On the other hand, there is no pair of arcs that covers the circle, so $\text{RATIO}(C) > 2$. We got a contradiction since $\text{RATIO}(I) = \text{RATIO}(C)$, therefore there exists an arc y_j that does not overlap x_i. $\square$

Observation 16. *For every arc y_j and two arcs x_i and x_{i+1}, if y_j covers $r(x_i)$ then y_j also covers $\ell(x_{i+1})$. Similarly, if y_j covers $\ell(x_{i+1})$ then y_j also covers $r(x_i)$.*

Proof. Both arcs y_j and x_{i+1} covers $r(x_i)$, therefore y_j and x_{i+1} overlap each other. Assume y_j overlaps the right side of both x_i and x_{i+1}. In this case, we can replace x_{i+1} with y_j in C and get a minimum bounding cycle that shares a vertex with I, contradicting Lemma 7. We prove the second claim symmetrically. $\square$

Observation 17. *For every arc x_i and two arcs y_j and y_{j+1}, if y_j covers $\ell(x_i)$ then y_{j+1} covers $r(x_i)$. Similarly, if y_{j+1} covers $r(x_i)$ then y_j covers $\ell(x_i)$.*

Proof. Assume that x_i does not intersect y_{j+1}, then we can replace y_j with x_i in I, contradicting Lemma 7. Since y_j covers $\ell(x_i)$ and is disjoint from y_{j+1}, we get that y_{j+1} covers $r(x_i)$. The second claim is proved symmetrically. $\square$

Observation 18. *The arc x_i overlaps y_j if and only if x_{i+1} overlaps y_{j+1}.*

Proof. Assume that the arc x_i overlaps y_j. If y_j covers $\ell(x_i)$ then y_{j+1} covers $r(x_i)$ by Observation 17, and therefore y_{j+1} covers $\ell(x_{i+1})$ by Observation 16. Otherwise, If y_j covers $r(x_i)$ then y_j covers $\ell(x_{i+1})$ by Observation 16 and therefore y_{j+1} covers $r(x_{i+1})$ by Observation 17. The other direction is proved symmetrically. $\square$

Observation 19. *For any arc x_i, let y_j be the first arc, to its right, which is disjoint from x_i out of the arcs of vertices of I. Then, y_j overlaps the right side of x_{i+1}. Symmetrically, for any arc x_i, the rightmost arc, which is disjoint from x_i, out of the arcs of vertices of I, overlaps the left side of x_{i-1}.*

Proof. Assume that y_j is disjoint from, x_{i+1}. Then, by Observation 18, the arc y_{j-1} does not overlap x_i. So, we can add x_i to I between y_{j-1} and y_j, and get a bounding co-cycle with a greater ratio, contradicting the fact that I is maximum. The second claim is proved symmetrically. □

Now we show how to construct the certificate of Theorem 14. Consider the arc x_1 in ϱ. This arc overlaps the right side of the arc x_0. So, by Observations 15 and 19 there is an arc y_b which is disjoint from x_0 and overlaps the right side of x_1. Also, by Observation 18 the arc y_{b+1} is disjoint from x_1. Symmetrically, the arc x_2 overlaps the right side of x_1, so, there is an arc y_a such that y_a is disjoint from x_2 and overlaps the left side of x_1. Also, the arc y_{a-1} is disjoint from x_1. Note that y_{a-1} and y_{b+1} may be the same arc. The arcs x_0 and y_a cover the same endpoint of x_1 so they overlap each other. Similarly, x_2 and y_b also overlap each other. Since I is a co-cycle, we may set y_0 to be y_a (using the analogous of Lemma 10 for co-cycles), we also set y_b to be y_r, with $r = b - a$. (See Fig. 6). We get that x_0y_0, x_0y_r, x_1y_0 and x_1y_r are all edges and that x_0y_r, x_1y_{p-1}, x_1y_{r+1} and x_2y_0 are all non-edges, these edges and non-edges are the certificate required by Theorem 14 for x_0. Using Observation 18, we get the certificate for every other $x_i \in C$.

We find y_a and y_b in $O(n)$ time, and from them we find the rest of the vertices that are involved in the certificate, and hence also all the edges and non-edges in it, in linear time. In total, we provide at most $10|C|$ edges and non-edges as a certificate that G is not a unit circular-arc graph (actually the number of different edges and non-edges is at most $8|C|$, because of duplicates), and keep the $O(n)$ size bound.

4.7. The authentication algorithm

If G is a unit circular-arc graph then the certificate is a unit circular-arc model. This certificate can be authenticated by verifying that it is a circular-arc model of G as in Section 3.4 and comparing the lengths of all arcs. This takes $O(n + m)$ time.

If G is not a proper circular-arc graph, then the certificate and its authentication algorithm are as in Section 3.4. The authentication in this case takes $O(n)$ time.

If G is a proper circular-arc and not a unit circular-arc graph, then the recognition algorithm provides a set of $O(n)$ edges and non-edges which by Theorem 14 proves that G is not a unit circular-arc graphs. Since the number of edges and non-edges in $O(n)$, it takes $O(n)$ to validate this certificate. We note that this does not prove that G is a proper circular-arc graph.

When the algorithm found that G is not a unit circular-arc graph, all possible certificates can be authenticated in $O(n)$. Therefore, the certificate for this case is a strong certificate when G is given by ordered adjacency lists.

5. Further work

A natural question that is left open is to find a certifying recognition algorithm for circular-arc graphs. The only known characterizations of circular-arc graphs are the ones of Tucker [22] and Gavril [5], but it is not clear how to derive from these characterizations a certificate for a graph which is not a circular-arc graph. So, the first step towards finding a certifying algorithm for circular-arc graphs would be finding a new characterization of circular-arc graphs from which we can derive a certificate when the graph is not a circular-arc graph.

Our certificate for a graph that is not a proper circular-arc graph is one of two possible kinds. It is either an odd length cycle in the incompatibility graph, or an induced subgraph of constant size. The characterizations that we use for proper circular-arc graphs, Theorems 1 and 3, do not provide one simple condition that fits all graphs. However, the characterization of Skrien [20] and the characterization of Deng, Hell and Huang [4] do provide such a condition. It is an open question whether it is possible to derive a certifying algorithm for proper circular-arc graph based on these or other characterizations that do not split into two cases. The number of different kinds of certificates even increases for unit circular-arc graphs, which is based on recognizing whether a proper circular-arc graph is a unit circular-arc graph, like other characterizations and recognition algorithms for unit circular-arc graphs [23,3,13].

For unit circular-arc graphs we use, in fact, two recognition algorithms, the algorithm of Lin and Szwarcfiter [13] and our algorithm in Section 4. The former is used to give a certificate when the input graph is a unit circular-arc graph and the latter is used to give a certificate otherwise. Therefore every unit circular-arc graph is recognized twice. It is an open problem to find a single algorithm that can provide both certificates.

References

[1] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, J. Comput. System Sci. 13 (3) (1976) 335–379.

[2] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph classes: A survey, in: SIAM Monographs on Discrete Mathematics and Applications 3, SIAM, Philadelphia, PA, 1999.

[3] G. Durán, A. Gravano, R.M. McConnell, J.P. Spinrad, A. Tucker, Polynomial time recognition of unit circular-arc graphs, J. Algorithms 58 (1) (2006) 67–78.

[4] X. Deng, P. Hell, J. Huang, Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs, SIAM J. Comput. 25 (2) (1996) 390–403.

[5] F. Gavril, Algorithms on circular-arc graphs, Networks 4 (4) (1974) 357–369.

[6] M.C. Golumbic, Algorithmic graph theory and perfect graphs, in: Computer Science and Applied Mathematics, Academic Press, New York, 1980.

[7] P. Hell, J. Huang, Certifying LexBFS recognition algorithms for proper interval graphs and proper interval bigraphs, SIAM J. Discrete Math. 18 (3) (2004) 554–570.

[8] P. Hell, J. Huang, Interval bigraphs and circular arc graphs, J. Graph Theory 46 (4) (2004) 313–327.
[9] W.-L. Hsu, R.M. McConnell, PC-trees and circular-ones arrangements, Theoret. Comput. Sci. 296 (1) (2003) 99–116.
[10] D. Kratsch, R.M. McConnell, K. Mehlhorn, J.P. Spinrad, Certifying algorithms for recognizing interval graphs and permutation graphs, SIAM J. Comput. 36 (2) (2006) 326–353.
[11] H. Kaplan, Y. Nussbaum, Certifying algorithms for recognizing proper circular-arc graphs and unit circular-arc graphs, in: 32nd International Workshop on Graph-Theoretic Concepts in Computer Science, WG, in: Lecture Notes in Computer Science, vol. 4271, 2006, pp. 289–300.
[12] H. Kaplan, Y. Nussbaum, A simpler linear-time recognition of circular-arc graphs, in: 10th Scandinavian Workshop on Algorithm Theory, SWAT, in: Lecture Notes in Computer Science, vol. 4059, 2006, pp. 41–52.
[13] M.C. Lin, J.L. Szwarcfiter, Unit circular-arc graph representations and feasible circulations, SIAM J. Discrete Math. 22 (1) (2008) 409–423.
[14] R.M. McConnell, Linear-time recognition of circular-arc graphs, Algorithmica 37 (2) (2003) 93–147.
[15] R.M. McConnell, A certifying algorithm for the consecutive-ones property, in: Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 2004, pp. 768–777.
[16] D. Meister, Recognition and computation of minimal triangulations for AT-free claw-free and co-comparability graphs, Discrete Appl. Math. 146 (3) (2005) 193–218.
[17] K. Mehlhorn, S. Näher, The LEDA Platform for Combinatorial and Geometric Computing, Cambridge University Press, 1999.
[18] H. Müller, Recognizing interval digraphs and interval bigraphs in polynomial time, Discrete Appl. Math. 78 (1–3) (1997) 189–205. Erratum: http://www.comp.leeds.ac.uk/hm/patch.ps.
[19] F.S. Roberts, Indifference graphs, in: Proof Techniques in Graph theory: Proceedings of the Second Ann Arbor Conference, 1969, pp. 139–146.
[20] 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) 309–316.
[21] J.P. Spinrad, Efficient graph representations, in: Fields Institute Monographs 19, American Mathematical Society, Providence, RI, 2003.
[22] A. Tucker, Matrix characterizations of circular-arc graphs, Pacific J. Math. 39 (2) (1971) 535–545.
[23] A. Tucker, Structure theorems for some classes of circular-arc graphs, Discrete Math. 7 (1974) 167–195.
[24] R.E. Tarjan, M. Yannakakis, Addendum: Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, SIAM J. Comput. 14 (1) (1985) 254–255.