SIAM J. DISC. MATH.
Vol. 7, No. 4, pp. 647–655, November 1994

012

TREEWIDTH OF CIRCULAR-ARC GRAPHS*

RAVI SUNDARAM†, KARAN SHER SINGH‡, AND C. PANDU RANGAN§

Abstract. The treewidth of a graph is one of the most important graph-theoretic parameters from the algorithmic point of view. However, computing the treewidth and constructing a corresponding tree-decomposition for a general graph is NP-complete. This paper presents an algorithm for computing the treewidth and constructing a corresponding tree-decomposition for circular-arc graphs in $O(n^3)$ time.

Key words. treewidth, circular-arc graphs, tree-decomposition, complexity

AMS subject classifications. 05C85, 68Q25, 68R10

1. Introduction.

The notion of the treewidth of a graph has a large number of applications in many areas, like algorithmic graph theory, VLSI design, and so forth [Ar85]. Recently there has been a growing interest in the study of the treewidth and the tree-decomposition of a graph from the algorithmic point of view.

Robertson and Seymour, in their classic series of papers on graph minors, introduced the notion of tree-decomposition and treewidth of a graph [RS86]. Most of their results are existential in nature. Several classes of problems that are NP-complete for an arbitrary graph admit polynomial-time solutions on graphs with bounded treewidth [ALS88]. Bodlaender [Bo88], [Bo89], [Bo93] has done extensive studies on the sequential and parallel algorithmic aspects of graphs with bounded treewidth.

Computing the treewidth and the corresponding tree-decomposition is known to be NP-complete for general graphs [ACP87]. For fixed k, the problem of determining whether the treewidth of a given graph is at most k can be solved in polynomial time with dynamic programming [ACP87], and in $O(n^2)$ time with graph minor theory [RS86]. The only known[1] algorithm for computing the treewidth of special classes of graphs is for cographs [BM90]. In this paper, we show that the treewidth of circular-arc graphs and the corresponding tree-decomposition can be found in $O(n^3)$ time. We do this by showing that among all possible tree-decompositions of the given circular-arc graph, only a small fraction need be considered to find one with minimum tree-width, namely, those corresponding to the planar triangulations of a certain circuit—and such tree-decompositions can be investigated quite easily with a polynomial-time algorithm.

The organization of the rest of the paper is as follows. In §2 basic definitions and preliminary results are given. Section 3 contains constructions, lemmas, and theorems that are essential to the proof of correctness of the algorithm. In §4 the actual algorithm and its analysis are presented. Section 5 contains concluding remarks.

---

* Received by the editors January 30, 1991; accepted for publication (in revised form) September 13, 1993. A preliminary version of this paper appeared in WADS'91. This research was conducted at the Indian Institute of Technology, Madras, as part of an ongoing project headed by C. Pandu Rangan.

† Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139 (koods@theory.lcs.mit.edu).

‡ Department of Computer and Information Science, Ohio State University, 2036 Neil Avenue, Columbus, Ohio 43210 (singh-k@cis.ohio-state.edu).

§ Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India 600036 (rangan@iitm.ernet.in).

[1] More results [BKK93], [HM92] in this direction have emerged since the appearance of this paper in a conference.

2. Definitions and Preliminary Results. If S is a set, then let $|S|$ denote the cardinality of S. Let $G = (V, E)$ be a graph. We also refer to the vertex set and edge set of G by $V(G)$ and $E(G)$, respectively.

A *tree-decomposition* of G is a pair $(\{X_i | i \in I\}, T = \{I, F\})$, with $\{X_i | i \in I\}$ a family of subsets of V, and T a tree with the following properties:

(1) $\bigcup_{i \in I} X_i = V$;

(2) $\forall_{(u,v) \in E} \exists_{i \in I} (u \in X_i) \bigwedge (v \in X_i)$;

(3) $\forall_{v \in V}$, the set $\{i | (i \in I) \bigwedge (v \in X_i)\}$ forms a connected subtree of T.

The *treewidth* of a tree-decomposition $(\{X_i | i \in I\}, T = \{I, F\})$, denoted by treewidth$((\{X_i | i \in I\}, T = \{I, F\}))$, is $\max_{i \in I}(|X_i| - 1)$. The treewidth of a graph G, denoted by treewidth(G), is the minimum treewidth of a tree-decomposition of G taken over all possible tree-decompositions of G.

A graph is said to be *chordal* [Go80] if it has no chordless cycle of size greater than 3. The treewidth of a graph G can be similarly defined to be one less than the smallest clique number of a chordal graph containing G.

A *circular-arc* graph [Go80], [SP89] is the intersection graph of arcs around the unit circle. Consider a unit circle with a fixed reference, say O, on it. Any point on the circumference of the circle can be uniquely identified by its distance from O along the circle, say, in the clockwise direction. Thus, we use the same symbol for the point as well as its distance from O. Let $AF = \{A_0, A_1, \ldots, A_{n-1}\}$ be a family of arcs on a unit circle, and let $G = (V, E)$, $|V| = n, |E| = m$, be the circular-arc graph with AF as its intersection model. The arc A_i is represented by the ordered pair $(l(A_i), r(A_i))$, where $l(A_i)$ and $r(A_i)$ denote its left and right end points, respectively. The arc A_i exists on the circle as a traversal in the clockwise direction from $l(A_i)$ to $r(A_i)$, along the circumference of the circle. Further, we assume that $l(A_i) \leq l(A_{i+1}), 0 \leq i \leq n-2$. We represent arcs in AF by uppercase letters and the corresponding vertices in G by lowercase letters. (See Figs. 1 and 2.)

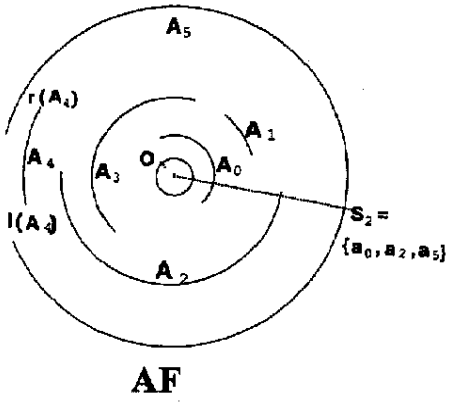

FIG. 1. *An intersection model.*

Unless otherwise stated, we assume $G = (V, E)$ to be a circular-arc graph with an arbitrary tree-decomposition $(\{X_v | v \in V(T)\}, T)$.

In general, the arc segment (x, y) exists on the circle as a traversal in the clockwise direction from x to y. (x, y) is said to *contain* position s if either $x \leq s \leq y$, or $(x \geq y) \bigwedge (s \geq x \bigvee s \leq y)$.

Every vertex $a_i \in G$ defines a *left clique* S_i, comprising the set of vertices $\{a_j | A_j$ contains $l(A_i)\}$. An *intersection clique* Q_i for the vertex a_i is the clique made

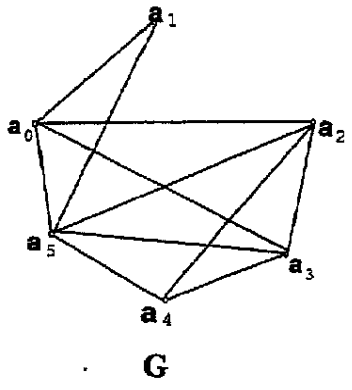

FIG. 2. *The circular-arc graph.*

up of the vertices in the set $\{a_j | a_j \in (S_i \bigcap S_{i+1})\}$. (See Fig. 3.).

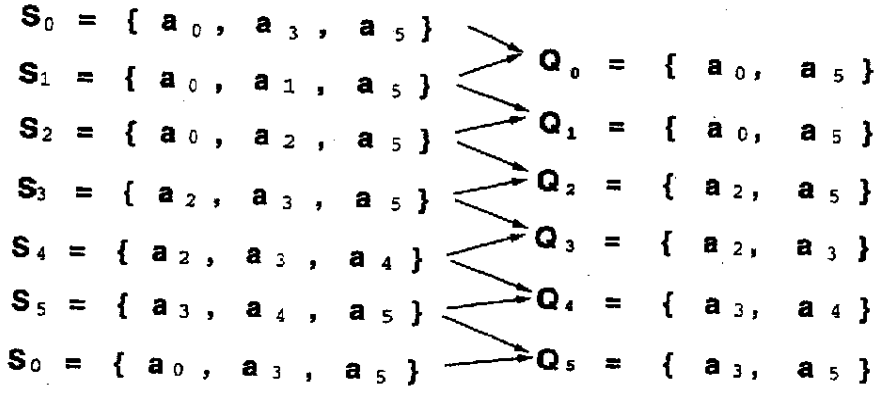

FIG. 3. *Corresponding left cliques and intersection cliques.*

Note. Unless otherwise mentioned, all indices are assumed to be modulo n. Let

$$VQ = \bigcup_{i=0}^{n-1} Q_i.$$

Let GQ be the subgraph of G induced by VQ.

For every $v \in V$, let $ST(v)$ be the subtree of T with vertices $\{i | (i \in V(T)) \bigwedge (v \in X_i)\}$ in the tree-decomposition $(\{X_i | i \in V(T)\}, T)$. Let $ST(Y) = \bigcap_{v \in Y} ST(v)$, where Y is a subset of V.

By definition, the vertices of $ST(Y)$ also form a connected subtree of T, although it may be empty.

LEMMA 2.1 (clique containment lemma.). *In any tree-decomposition of G, for any clique K of G, $ST(K)$ is nonempty* [Bo88], [BM90].

LEMMA 2.2. *The intersection graph of a set of subtrees of a tree is chordal* [Go80].

LEMMA 2.3. *Given a cycle C of n vertices, the set of minimal chordal graphs, which contain C as a subgraph, is equal to the set of planar triangulations of C with $n-3$ diagonal edges.*

LEMMA 2.4. *Any vertex v of G is contained in a nonempty consecutive subset $S_i, S_{i+1}, \ldots, S_j$ of the set of left cliques for some i and j, and in a consecutive subset $Q_i, Q_{i+1}, \ldots, Q_{j-1}$ of the set of intersection cliques, which is empty if $i = j$.*

LEMMA 2.5. *In any tree-decomposition of the circular-arc graph G, $ST(S_i)$ and $ST(S_{i+1})$ are subtrees of $ST(Q_i)$.*

Further, we assume no Q_i is empty, otherwise the graph reduces to an interval graph whose treewidth is trivial to compute.

COROLLARY 2.6. *$ST(Q_i) \bigcap ST(Q_{i+1})$ is nonempty for any tree-decomposition of G.*

LEMMA 2.7. *Let G be a graph. Let v be a vertex in G with degree d whose neighbours form a clique. Then* $\text{treewidth}(G) = \max(\text{treewidth}(G - v), d)$.

Proof. It is obvious that $\text{treewidth}(G) \geq \max(\text{treewidth}(G - v), d)$. To prove the upper bound, consider any chordal graph containing $G - v$; adding v yields a chordal graph containing G. (Recall that the treewidth of G is one less than the smallest clique number of a chordal graph containing G.) □

COROLLARY 2.8. $\text{Treewidth}(G) = \max(\max_{i \in I}(|S_i| - 1), \text{treewidth}(GQ))$.

Proof. From the definition of treewidth, it follows that $\text{treewidth}(G) \geq \max(\max_{i \in I}(|S_i| - 1), \text{treewidth}(GQ))$. The upper bound follows from Lemma 2.7 and the definition of GQ. □

By corollary 2.8, we see that computing $\text{treewidth}(GQ)$ is the key to computing the treewidth of G. In what follows, we are mainly concerned with the treewidth of GQ. Note that GQ is also a circular-arc graph.

Let $G' = (V', E') = IGST(\{X_v | v \in V(T)\}, T)$ be the intersection graph of the subtrees $ST(Q_i), 0 \leq i \leq n-1$, for a tree-decomposition $(\{X_v | v \in V(T)\}, T)$ of the circular-arc graph GQ. Let $V' = v'_0, v'_1, \ldots, v'_{n-1}$, where v'_i is the vertex corresponding to $ST(Q_i)$. From corollary 2.6, we see that $(v'_i, v'_{i+1}) \in E'$. Let CYQ be the cycle consisting of the vertices $v'_0, v'_1, \ldots, v'_{n-1}$, in that order. CYQ may also be referred to as the cycle corresponding to the intersection cliques.

LEMMA 2.9. *For any tree-decomposition $(\{X_v | v \in V(T)\}, T)$ of the circular-arc graph GQ, $IGST(\{X_v | v \in V(T)\}, T)$ (1) is chordal, and (2) contains CYQ as a subgraph.*

Proof. Part (1) follows from Lemma 2.2. (2) follows from Corollary 2.6. □

3. Constructions and Results.

Construction 1. For a given planar triangulation of CYQ, construct a tree-decomposition $(\{X_v | v \in V(TQ)\}, TQ)$ of GQ as follows: Generate TQ to equal the dual of the planar triangulation without the vertex corresponding to the external face. Let v be any vertex of TQ and v'_i, v'_j, and v'_k be the vertices of the triangular face whose dual is the vertex v. Now, let $X_v = Q_i \bigcup Q_j \bigcup Q_k$ be the set associated with the vertex v. For each $v \in V(TQ)$ generate X_v. (See Fig. 4.)

Note. Henceforth, we assume the absence of a vertex corresponding to the external face in all references to the dual of a (planar) graph.

THEOREM 3.1. *Construction* 1 *generates a tree-decomposition of GQ.*

Proof. It is easy to see that TQ is indeed a tree.

(i) We prove that $\bigcup_{v \in TQ} X_v = VQ$. For any vertex $a_i \in VQ$, $a_i \in Q_i$, and since v'_i occurs in some triangular face, the corresponding dual vertex v has an associated set X_v that contains Q_i, and hence a_i. Therefore, for every vertex $a_i \in VQ$, there exists a $v \in V(TQ)$ such that $a_i \in X_v$. Further, by construction, $\forall_{v \in V(TQ)} X_v \subseteq VQ$.

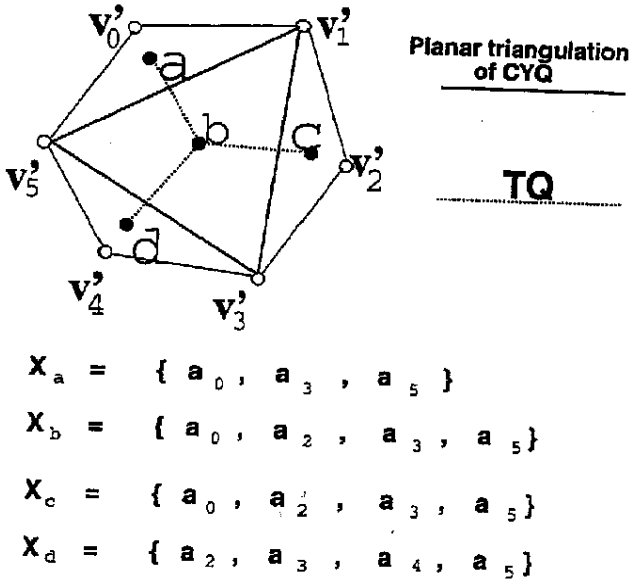

FIG. 4. *Construction* 1.

Therefore,

(1) $$\bigcup_{v \in TQ} X_v = VQ.$$

(ii) We prove that $\forall_{(a_k,a_l)\in E(GQ)} \exists_{w\in V(TQ)}(a_k \in X_w) \bigwedge (a_l \in X_w)$. For any edge $(a_k, a_l) \in E(GQ), (a_k \in S_k) \bigwedge (a_l \in S_k)$, or $(a_k \in S_l) \bigwedge (a_l \in S_l)$, say $a_k, a_l \in S_k$, without loss of generality. Then $a_k \in Q_k$ and $a_l \in Q_{k-1}$. Since v'_k and v'_{k-1} are adjacent in CYQ, they occur in some triangular face, say, one whose corresponding dual vertex is w. Then w has an associated set X_w that contains Q_k and Q_{k-1}, and therefore both a_k and a_l. Therefore,

(2) $$\forall_{(a_k,a_l)\in E(GQ)} \exists_{w\in V(TQ)}(a_k \in X_w) \wedge (a_l \in X_w).$$

(iii) We prove that $\forall_{a_i \in VQ}, \{v | (v \in V(TQ)) \bigwedge (a_i \in X_v)\}$ forms a connected subtree of TQ. By Lemma 2.4, any vertex a_i of GQ occurs in a consecutive subset $Q_i, Q_{i+1}, \ldots, Q_{j-1}$ of the intersection cliques. Let a triangular face be said to intersect a set of vertices if at least one of the three vertices of the face belongs to the set. Thus we must prove that, given a planar triangulation of a cycle and a segment of the cycle, the set of vertices in the dual, whose corresponding triangular faces intersect the segment, forms a connected subtree (in the dual). The proof follows by induction on the number of vertices in the segment. If the segment contains exactly one vertex, then the corresponding subtree is simply a path in the dual tree. Assume that it holds for all segments containing k vertices. Consider a segment containing $k+1$ vertices, numbered 1 to $k+1$ in order along the cycle. Consider the subsegment of k vertices formed by dropping one of the end vertices, say the $k+1$th. Corresponding to the subsegment, we have a connected subtree in the dual (by the induction hypothesis), and corresponding to the dropped end vertex, we have a path that intersects with the subtree (the intersection contains at least the dual (vertex) of the common triangular

face of the kth and $k+1$th vertices in the segment). Since the dual (TQ) is a tree, we obtain a connected subtree in the dual that is the union of the subtree corresponding to the subsegment of k vertices and the path corresponding to the one end vertex of the segment. Therefore,

$$\forall_{a_i \in VQ}, \{v | (v \in V(TQ)) \wedge (a_i \in X_v)\} \tag{3}$$

is a connected subtree of TQ.

From (1), (2), and (3) we conclude that Construction 1 generates a tree-decomposition of GQ. □

Consider any tree-decomposition $(\{X_v | v \in V(T')\}, T')$ of GQ such that $G' = IGST(\{X_v | v \in V(T')\}, T')$, the intersection graph of the subtrees corresponding to the intersection cliques, is not a planar triangulation of CYQ. By Lemma 2.9, G' is chordal and contains CYQ. Consider any minimal subgraph SG' of G' that contains CYQ and is chordal (obviously, SG' is an edge-induced subgraph). By Lemma 2.3, SG' must be a planar triangulation of CYQ. Let $(\{X_v | v \in V(TQ')\}, TQ')$ be the tree-decomposition corresponding to SG' obtained by Construction 1.

THEOREM 3.2. *Treewidth*$(\{X_v | v \in V(TQ')\}, TQ') \leq$ *treewidth*$(\{X_v | v \in V(T')\}, T')$.

Proof. By theorem 3.1, $(\{X_v | v \in V(TQ')\}, TQ')$ is a tree-decomposition of GQ. To prove the theorem, it is sufficient to show that $\forall_{u \in V(TQ')} \exists_{v \in V(T')} : X_u \subseteq X_v$.

Consider any $u \in V(TQ')$. By Construction 1, $X_u = Q_i \bigcup Q_j \bigcup Q_k$ for some i, j, and k. (Remember that in CYQ, there is a vertex v_i' for each Q_i.) Hence, SG' contains the edges (v_i', v_j'), (v_j', v_k'), and (v_i', v_k'). Since SG' is only a subgraph of G', these edges are also present in G'. This implies that the sets $ST(Q_i)$, $ST(Q_j)$, and $ST(Q_k)$ pairwise overlap, but because these are subtrees, they have a nonempty intersection in T', i.e., $ST(Q_i) \bigcap ST(Q_j) \bigcap ST(Q_k)$ in T' is nonempty. Consider some vertex $v \in V(T')$ in this nonempty intersection. X_v contains Q_i, Q_j and Q_k. Therefore, $X_v \supseteq X_u$.

Hence, it has been proved that treewidth$(\{X_v | v \in V(TQ')\}, TQ') \leq$ treewidth$(\{X_v | v \in V(T')\}, T')$. □

COROLLARY 3.3. *The treewidth of GQ is equal to the minimum treewidth over all tree-decompositions $(\{X_v | v \in V(TQ)\}, TQ)$ of GQ that satisfy the following: $G' = IGST(\{X_v | v \in V(TQ)\}, TQ)$ the corresponding intersection graph of the intersection cliques, contains CYQ and is a planar triangulation of CYQ.*

Proof. It follows from Theorems 3.1 and 3.2, and the fact that $IGST(\{X_v | v \in V(T)\}, T)$ is chordal and contains CYQ for any tree-decomposition $(\{X_v | v \in V(T)\}, T)$ of GQ. □

Construction 2. Given any tree-decomposition $(\{X_v | v \in V(TQ)\}, TQ)$ of GQ such that $G' = IGST(\{X_v | v \in V(TQ)\}, TQ)$ contains CYQ and is chordal, construct a tree-decomposition of G as follows: For each $a_i \in V$, add a new vertex b_i to TQ; attach b_i by an edge to any one vertex $v \in V(TQ)$ such that X_v contains Q_{i-1} and Q_i; and set $X_{b_i} = S_i$. Let $(\{X_v | v \in V(TQ_G)\}, TQ_G)$ denote the resulting tree-decomposition. (See Fig. 5.).

It is easy to see that Construction 2 is well defined and that it generates a tree-decomposition of G. Further, by Corollary 2.8, if the tree decomposition $(\{X_v | v \in V(TQ)\}, TQ)$ achieves the minimum tree-width for GQ, then the treewidth of the tree-decomposition $(\{X_v | v \in V(TQ_G)\}, TQ_G)$ equals the treewidth of G.

4. Algorithm and its analysis. The basic algorithm now boils down to computing treewidth(GQ), and a corresponding tree-decomposition. The problem of com-

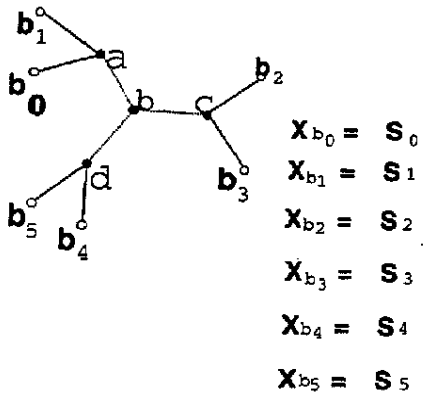

FIG. 5. *Construction* 2.

puting treewidth(GQ) can be rephrased as follows.

Given an n-gon $v_0, v_1, \ldots, v_{n-1}$, and sets A_i associated with each vertex v_i such that every element occurs in a consecutive subset $A_i, A_{i+1}, \ldots, A_j$, define the cost of a triangle with vertices v_i, v_j, and v_k to be $|A_i \bigcup A_j \bigcup A_k| - 1$. Define the cost of a planar triangulation to be the maximum cost over all triangular faces in the triangulation. The problem (restated) involves finding the minimum cost over all planar triangulations, and the corresponding planar triangulation (refer to Corollary 3.3).

Specifically, in this case it is required to find the minimum-cost planar triangulation of the n-gon $v'_0, v'_1, \ldots, v'_{n-1}$, with the set associated with v'_i being Q_i. This is done by a dynamic programming approach. $TWCYQ_j(i)$ is defined to be the minimum cost over all planar triangulations of the subpolygon $v'_i, v'_{i+1}, \ldots, v'_{i+j-1}$. Therefore, treewidth($GQ$) = $TWCYQ_n(0)$.

ALGORITHM.

Input $(l(A_i), r(A_i))$ representation of the vertices a_i of the circular-arc graph G.

Steps

1. Construct $S_i, 0 \leq i \leq n-1$ (in sorted order).
2. Construct $Q_i, 0 \leq i \leq n-1$ (in sorted order).
3. Compute treewidth(GQ) = $TWCYQ_n(0)$ using dynamic programming, and find a corresponding planar triangulation.
4. Employ Construction 1 on the planar triangulation to obtain the tree-decomposition of GQ.
5. Employ Construction 2 (using the tree-decomposition of GQ obtained from step 4) to obtain the tree-decomposition of G and compute the corresponding treewidth.

Output Treewidth(G) and corresponding tree-decomposition.

Details of step 3.

3.1. For all $i, 0 < i < n-1$, compute

$$TWCYQ_1(i) = TWCYQ_2(i) = 0,$$

$$TWCYQ_3(i) = |Q_i \cup Q_{i+1} \cup Q_{i+2}| - 1.$$

3.2. For $j = 4$ to n, and
for $i = 0$ to $n-1$, compute

$$TWCYQ_j(i) = \min_{k=1}^{j-2}\{\max(TWCYQ_{k+1}(i), TWCYQ_{j-k}(i+k), |Q_i \cup Q_{i+k} \cup Q_{i+j-1}| - 1)\}.$$

To actually find the corresponding planar triangulation, pointers and additional book-keeping, information could be stored so that, by retracing, the optimum triangulation can be constructed.

Proof of correctness. It follows from Theorems 3.1 and 3.2 and Corollary 3.3. □

Time complexity. The dynamic programming employed in step 3 itself takes $O(n^3)$ time. But the bottleneck is finding all the $|Q_i \bigcup Q_{i+k} \bigcup Q_{i+j-1}|$. Since the union in the innermost loop takes $O(n)$ time, naively it would seem that step 3 takes $O(n^4)$ time. However, finding all the $|Q_i \bigcup Q_{i+k} \bigcup Q_{i+j-1}|$ can be done faster than $O(n^4)$. Note that the sets Q_i and the elements they contain can be represented as a circular list of events. The events represent the start of an element, the end of an element, and a set. An element is considered to belong to all those sets whose event lies in the clockwise traversal from the start event, to the end event of this element. To show that all the $|Q_i \bigcup Q_{i+k} \bigcup Q_{i+j-1}|$ can be computed in $O(n^3)$ time, all we must show is that for any fixed set B, all the $|B \bigcup Q_i|$ can be computed in $O(n)$ time. This is done as follows: Remove all events corresponding to elements in B from the list, then do a sweep around the circle maintaining a counter that represents the arcs of $Q_i - B$ covering the point you are at, decrement the counter when you hit the end of an arc, and increment it at the start of an arc. When you arrive at the event corresponding to a set, output the counter plus $|B|$.

5. Final remarks. In this paper, we presented an algorithm to compute the treewidth of a circular-arc graph in $O(n^3)$ time. Since our algorithm simply uses straightforward dynamic programming, we conjecture that it is possible to improve the time complexity substantially. We note that there is an $O(n \log n)$ algorithm for the following problem (which is syntactically very similar to the one we solve).

Given an n-gon $v_0, v_1, \ldots, v_{n-1}$, and numbers A_i associated with each vertex v_i, define the cost of a triangle with vertices v_i, v_j and v_k to be $A_i \times A_j \times A_k$. Define the cost of a planar triangulation to be the sum over all triangular faces in the triangulation. The problem involves finding the minimum cost over all planar triangulations and the corresponding planar triangulation.

The fastest known algorithm for this problem was the straightforward $O(n^3)$ dynamic programming algorithm, until Hu and Shing devised an extremely clever $O(n \log n)$ algorithm [HS80], [Ya80]. So it is possible that a closer analysis of the underlying problem will yield a faster algorithm. Another interesting direction is to develop an algorithm to compute the treewidth for a substantially larger class of graphs.

Acknowledgments. The authors thank the anonymous referees for suggesting the many improvements and corrections that have gone a long way toward improving the presentation of this paper.

REFERENCES

[Ar85] S. ARNBORG, *Efficient algorithms for combinatorial problems on graphs with bounded decomposability—A survey*, BIT, 25 (1985), pp. 2–23.

[ACP87] S. ARNBORG, D. G. CORNEIL, AND A. PROSKUROWSKI, *Complexity of finding embeddings in a k-tree*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 277–284.

[ALS88] S. ARNBORG, J. LAGERGREN, AND D. SEESE, *Problems easy for tree-decomposable graphs*, in Proc. 15th ICALP, LNCS 317 (1988), pp. 38–51.

[Bo88] H. L. BODLAENDER, *NC-algorithms for graphs with small treewidth*, in Proc. WG'88, LNCS 344 (1988), pp. 1–10.

[Bo89] ———, *On linear time minor tests and depth first search*, In Proc. WADS'89, LNCS 382 (1989), pp. 577–590.

[Bo93] ———, *A linear time algorithm for finding tree-decompositions of small treewidth*, in Proc. 25th Annual ACM Symposium on the Theory of Computing, 1993, pp. 226–234.

[BM90] H. L. BODLAENDER AND R. H. MOHRING, *The pathwidth and treewidth of cographs*, in Proc. SWAT'90, LNCS 447 (1990), pp. 301–309.

[BKK93] H. L. BODLAENDER, T. KLOKS AND D. KRATSCH, *Treewidth and pathwidth of permutation graphs*, in Proc. 20th ICALP, 1993, to appear.

[Go80] M. C. GOLUMBIC, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[HM92] M. HABIB AND R. H. MOHRING, *Treewidth of cocomparability graphs and a new order-theoretic parameter*, Tech. Report 336/1992, Technische Univ. Berlin, 1992.

[HS80] T. C. HU AND M. T. SHING, *Some theorems about matrix multiplication* in Proc. 21st Annual IEEE Symposium on Foundations of Computer Science, 1980, pp. 28–35.

[RS86] N. ROBERTSON AND P. SEYMOUR, *Graph minors. II. Algorithmic aspects of treewidth*, J. Algorithms, 7 (1986), pp. 309–322.

[SP89] A. SRINIVASA RAO AND C. PANDU RANGAN, *Linear algorithms for parity path and two path problems on circular-arc graphs*, in Proc. WADS'89, LNCS 382 (1989), pp. 267–290.

[Ya80] F. F. YAO, *Efficient dynamic programming using quadrangle inequalities*, in Proc. 12th Annual ACM Symposium on the Theory of Computing, 1980, pp. 429–435.