ALGORITHMS FOR THE MAXIMUM INDUCED BIPARTITE SUBGRAPH PROBLEM ON INTERVAL AND CIRCULAR-ARC GRAPHS

by

Rachel Manber
Giri Narasimhan

Computer Sciences Technical Report #696

April 1987

ALGORITHMS FOR THE MAXIMUM INDUCED BIPARTITE SUBGRAPH PROBLEM ON INTERVAL AND CIRCULAR-ARC GRAPHS

Rachel Manber* Giri Narasimhan*

Computer Sciences Department
University of Wisconsin-Madison
Madison, Wisconsin-53706.

April 1987

ABSTRACT

The **Maximum Induced Bipartite Subgraph Problem** (*MIBS*) is the problem of finding an induced bipartite subgraph with the largest number of vertices. The problem is known to be NP-complete for general graphs [LY80]. We present algorithms to find the maximum induced bipartite subgraph for interval graphs and for circular-arc graphs. These algorithms run in times $O(n \log n)$ and $O(n^3)$ respectively.

* Research Supported By A Grant From Wisconsin Alumni Research Foundation.

1. INTRODUCTION

Let $G = (V, E)$ be a **simple graph**, i.e., a finite, undirected, loopless graph without multiple edges. Let $\mathcal{F}$ be a family of nonempty sets. The graph G is called an **Intersection Graph for** $\mathcal{F}$ if there is a one-to-one correspondence between the vertices of G and the sets in $\mathcal{F}$ such that two vertices are adjacent iff their corresponding sets intersect. The set $\mathcal{F}$ forms an **Intersection Model** for the graph. Every graph is an intersection graph for some family of sets [Ma45]. We focus here on two special classes of intersection graphs. One class consists of **Interval Graphs** which are the intersection graphs for a family of intervals on a linearly ordered set. The other class consists of **Circular-Arc Graphs** which are intersection graphs for a family of intervals that lie on a circle (circular arcs). These two classes of graphs arise in practical applications including scheduling and compiler optimization [Go80].

The **Maximum Induced Bipartite Subgraph Problem** (*MIBS*) is the problem of finding an induced bipartite subgraph with the largest number of vertices. The problem is known to be NP-complete for general graphs [LY80]. We present here algorithms to find the maximum induced bipartite subgraph for interval graphs and for circular-arc graphs. These algorithms run in times $O(n \log n)$ and $O(n^3)$ respectively. The algorithms assume that their inputs are intersection models for the given graphs. An intersection model for interval graphs can be computed in $O(n + m)$ time [BL76]. An intersection model for circular-arc graphs can be found in $O(n^3)$ [Tuc80].

The maximum induced bipartite subgraph problem can be viewed as the problem of coloring the maximum number of vertices using 2 colors. There is a wide spectrum of problems of the same general type. These problems ask for the maximum number of vertices of a graph that can be colored using k colors. For instance, the problem of finding the largest independent set in a graph can be formulated as the problem of coloring the maximum number of vertices with 1 color. At the other end of this spectrum lies the general coloring problem. There are many variations and applications of the coloring problem, and they are usually NP-Complete. In particular, one can ask for the largest induced subgraph which is k-colorable for arbitrary values of k or just for small fixed values of k. We concentrate on the case $k = 2$.

For interval graphs, many of these coloring problems are known to be solvable in polynomial time. These include the maximum independent set problem [GLL82], the general coloring problem, and the problem of deciding whether, for a fixed k, the chromatic number of G is at most k [GLL82]. However, for circular-arc graphs the general coloring problem is NP-Complete, while the other problems mentioned above are known to be solvable in polynomial time [Ga74] [GJMP80]. Sections 2 and 3 contain polynomial-time algorithms for the Maximum Induced Bipartite Subgraph problem for Interval graphs and for Circular-arc graphs respectively.

2. INTERVAL GRAPHS

Let $I = \{u_i = (a_i, b_i) : i = 1, \ldots, n\}$ be an intersection model for the interval graph $G(I) = (V, E)$. That is, there is a one-to-one correspondence between the intervals in I and the vertices of G such that $[v_i, v_j] \in E$ if and only if the corresponding intervals u_i and u_j have a nonempty intersection. We note that there is no loss of generality in assuming that all intervals in I are open and share no endpoints.

Interval graphs are chordal. That is, every cycle of length greater than 3 has a chord. It follows that every bipartite interval graph is acyclic. Moreover, since every induced subgraph of an interval graph is an interval graph, every induced bipartite subgraph of an interval graph is acyclic.

Algorithm $\mathcal{A}$ below finds a maximum induced bipartite subgraph of a given interval graph. It first sorts the n right endpoints of the given n intervals and then scans the intervals in that order. The scanning process repeatedly chooses the two intervals with the leftmost right endpoints to be in the solution and deletes the two intervals as well as all intervals which intersects both of them. The scanning process can be done in linear time and hence the time complexity of Algorithm A is dominated by the time complexity of sorting, i.e., $O(n \log n)$.

Algorithm $\mathcal{A}$

INPUT: A set of intervals I given as a set of ordered pairs of endpoints.
OUTPUT: $I' \subseteq I$ such that the subgraph $G(I')$ induced by the intervals in I' is the largest induced bipartite subgraph in $G(I)$.

1. Sort the n right endpoints of the n intervals.
2. Let y_0 be the interval with the leftmost right endpoint.
3. $I' \longleftarrow \{y_0\}$
4. $I \longleftarrow I - \{y_0\}$.
5. **while** I is nonempty **do**
    6. Let y_1 be the interval in I with the leftmost right endpoint.
    7. Add y_1 to I'.
    8. $I \longleftarrow I - \{y_1\}$.
    9. If y_0 intersects y_1 then remove from I all intervals that intersect both y_0 and y_1.
    10. $y_0 \longleftarrow y_1$

    endwhile
11. Output I'.

The main idea behind Algorithm $\mathcal{A}$ is that there exists an optimal solution that contains the two intervals with the leftmost right endpoints. This fact is established in the next two lemmas. Theorem 3 proves the correctness of Algorithm $\mathcal{A}$.

Lemma 1: Let I be a family of intervals on a linearly ordered set, and let y_0 be the interval in I with the leftmost right endpoint. Then there exists a solution to the *MIBS* problem for $G(I)$ that contains y_0.

Proof: Let J be any solution to the *MIBS* problem. Suppose J does not include y_0. Let y_1 be the interval in J with the leftmost right endpoint. The interval y_1 must intersect

y_0 or else $J \cup \{y_0\}$ would be a solution of size larger than the size of J. We claim that $J' = J - \{y_1\} \cup \{y_0\}$ is a solution to the *MIBS* problem for $G(I)$. It suffices to show that $G(J')$ is bipartite. Let v_0 and v_1 be the vertices corresponding to intervals y_0 and y_1 respectively. If v is a vertex in $G(J)$ and $(v, v_1) \notin E$, then the interval corresponding to v must lie wholly to the right of y_1 and consequently must lie wholly to the right of y_0. It follows that $(v, v_0) \notin E$, and since $G(J)$ is bipartite so is $G(J')$.∎

Lemma 2: Let I be a family of intervals on a linearly ordered set and let y_0 and y_1 be the two intervals with the leftmost right endpoints in I. Then there exists a solution to the *MIBS* problem for $G(I)$ that contains both y_0 and y_1.

Proof: By Lemma 1 there exists a solution J' that includes y_0. Suppose $y_1 \notin J'$. Let V_1 and V_2 be a bipartition of the vertex set of $B(J')$. Let J_1' and J_2' be a partition of J' that corresponds to the vertex sets V_1 and V_2. We may assume that $y_0 \in J_1'$. We want to replace an appropriate element of J' by y_1 thus constructing a solution that contains both y_0 and y_1. Let y_2 be the interval with the leftmost right endpoint in J_2' and let $J_2 = J_2' - \{y_2\} \cup \{y_1\}$ and $J_1 = J_1'$. All the intervals in $J_2' - \{y_2\}$ lie entirely to the right of y_2 and hence must lie entirely to the right of y_1. It follows that the intervals in J_2' do not intersect each other and the intervals in $J = J_1 \cup J_2$ induce a bipartite subgraph of cardinality $|J'|$. Thus J is a solution to the *MIBS* problem for $G(I)$ that contains both y_0 and y_1.∎

Theorem 3: Let I be a family of intervals on a linearly ordered set. The output of Algorithm $\mathcal{A}$ on input I is a solution to the *MIBS* problem for $G(I)$.

Proof: We use induction on the number n of intervals in I. The claim is trivially true for $n = 1$. Assume that the algorithm works correctly on all sets with less than n intervals and let $|I| = n$.

Let S' be the solution found by Algorithm $\mathcal{A}$ on input I. Then S' is clearly bipartite. Denote by y_0 and y_1 the two intervals in I with the leftmost right endpoints. If y_0 and y_1

intersect, we define Δ as the set of intervals that intersect both of them, otherwise $\Delta = \emptyset$. Clearly $y_0, y_1 \in S'$. Moreover, Algorithm $\mathcal{A}$ will return the solution $S' - \{y_0\}$ on the input $I' = I - \{y_0\} - \Delta$. By the inductive assumption $S' - \{y_0\}$ is a correct solution for I'.

Let S be any solution on input I containing y_0 and y_1. By Lemma 2 such a solution exists. Next we show that $S - \{y_0\}$ is a solution on input I'. This would complete the proof of the Theorem since it implies that $|S| = |S'|$.

Clearly $S - \{y_0\}$ induces a bipartite subgraph in $G(I')$. Assume that $S - \{y_0\}$ is not a maximum size solution on input I'. Let T' be a solution on input I' such that $|T'| > |S - \{y_0\}|$. By Lemma 1, we can assume that $y_1 \in T'$. Let $T = T' \cup \{y_0\}$. We claim that the intervals in T correspond to a bipartite subgraph of $G(I)$. Indeed, if any triangle is formed in $G(T)$, it must have y_0 as one of its vertices. But, if an interval intersects y_0, it also intersects y_1, because the right endpoint of y_0 lies to the left of the right endpoint of y_1. Hence any triangles in $G(T)$ must be formed by the vertices corresponding to y_0, y_1 and one other interval. But all such intervals are in Δ, and $\Delta \cap I' = \emptyset$. It follows that T is a solution on input I. The fact that $|T| > |S|$ contradicts the assumption that S is a solution on input I. ∎

We conclude this section with the following simple observation on the nature of Algorithm $\mathcal{A}$. The rightmost right endpoint of a set of intervals S is called Right-end(S).

Lemma 4: Let I be a set of intervals on a straight line. When Algorithm $\mathcal{A}$ is applied to input I it returns a solution S such that Right-end(S) is the least (i.e., leftmost) among all solutions.

Proof: Let $u_l = (x_l, y_l)$ be the interval in S with the rightmost right endpoint. Consider any solution S' on input I. Let $u'_l = (x'_l, y'_l)$ be the interval with the rightmost right endpoint in S'. Since Algorithm $\mathcal{A}$ is correct, $|S| = |S'|$. Suppose $y'_l < y_l$. Now let R be the set of all intervals in I with right endpoints to the right of y'_l. Then $u_l \in R$ and $R \cap S' = \emptyset$. Also, let $T = R \cap S$. Since $u_l \in T$, we have $|T| \geq 1$. Clearly, S' is a solution on input $I - R$. However, on input $I - R$ Algorithm $\mathcal{A}$ will find the solution $S - T$, which is of suboptimal cardinality ($< |S'|$), thus contradicting the correctness of Algorithm $\mathcal{A}$. ∎

3. CIRCULAR-ARC GRAPHS

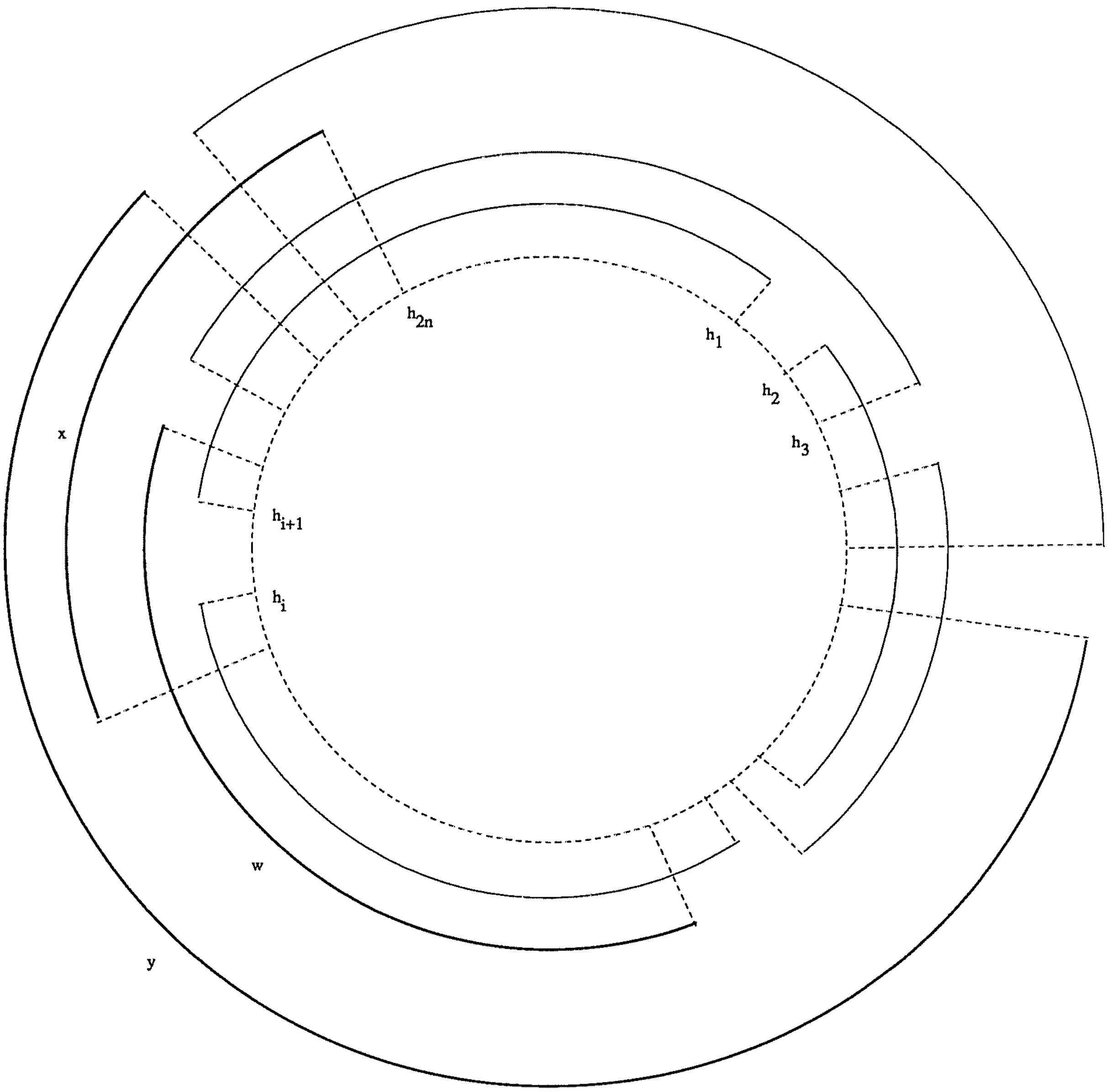
In this section we present an algorithm for solving the maximum induced bipartite problem for circular-arc graphs. The input for the algorithm is a collection $M = \{u_i = (x_i, y_i) : i = 1, \ldots, n\}$ of intervals on a circle (taken in the clockwise direction). The circular-arc graph corresponding to M is denoted by $G(M)$. Let $h_1, h_2, \ldots, h_{2n}, h_{2n+1} = h_1$, be an ordering of the endpoints of the circular arcs in the clockwise direction on the circle. Denote the interval (h_i, h_{i+1}) by a_i for $1 \leq i \leq 2n$, Let W_i be the set of all arcs in M that contain interval a_i and let $M_i = M - W_i$. See Figure 1.

Let S be a collection of circular arcs such that $G(S)$ is a solution to the *MIBS* Problem for $G(M)$. Since the intervals corresponding to an independent set in $G(M)$ cannot cover the entire circle, there exists an interval a_{i_0} such that W_{i_0} has at most one interval from S. If there exists an interval a_{i_0} such that $W_{i_0} \cap S = \emptyset$, then finding a maximum bipartite subgraph for $G(I)$ is equivalent to finding one for the interval graph $G(M_{i_0})$. Otherwise, there exists an interval a_{i_0} such that $|W_{i_0} \cap S| = 1$. In that case, finding a maximum bipartite subgraph for $G(I)$ is equivalent to finding one for $G(J)$, where $J = M_{i_0} \cup \{w\}$ for some $w \in W_{i_0}$. Note that $G(J)$ is a special kind of circular-arc graph which we call an **Elementary Circular-Arc Graph**. More precisely, an elementary circular-arc graph is a pair [$G(J)$,w] such that $G(J)$ is a circular-arc graph, w is in J, and $G(J - \{w\})$ is an interval graph. The arc w is called here the **Circular Interval**. See Figure 2.

Thus the *MIBS* problem for circular-arc graphs can be reduced to a set of sub-problems for interval graphs and for elementary circular-arc graphs.

Algorithm $\mathcal{B}$ below extends Algorithm $\mathcal{A}$ to work on elementary circular-arc graphs. It takes as input an intersection model of an elementary circular-arc graph $[G(I), w]$ and finds a solution S of maximum cardinality such that $G(S)$ is an induced bipartite subgraph and $w \in S$.

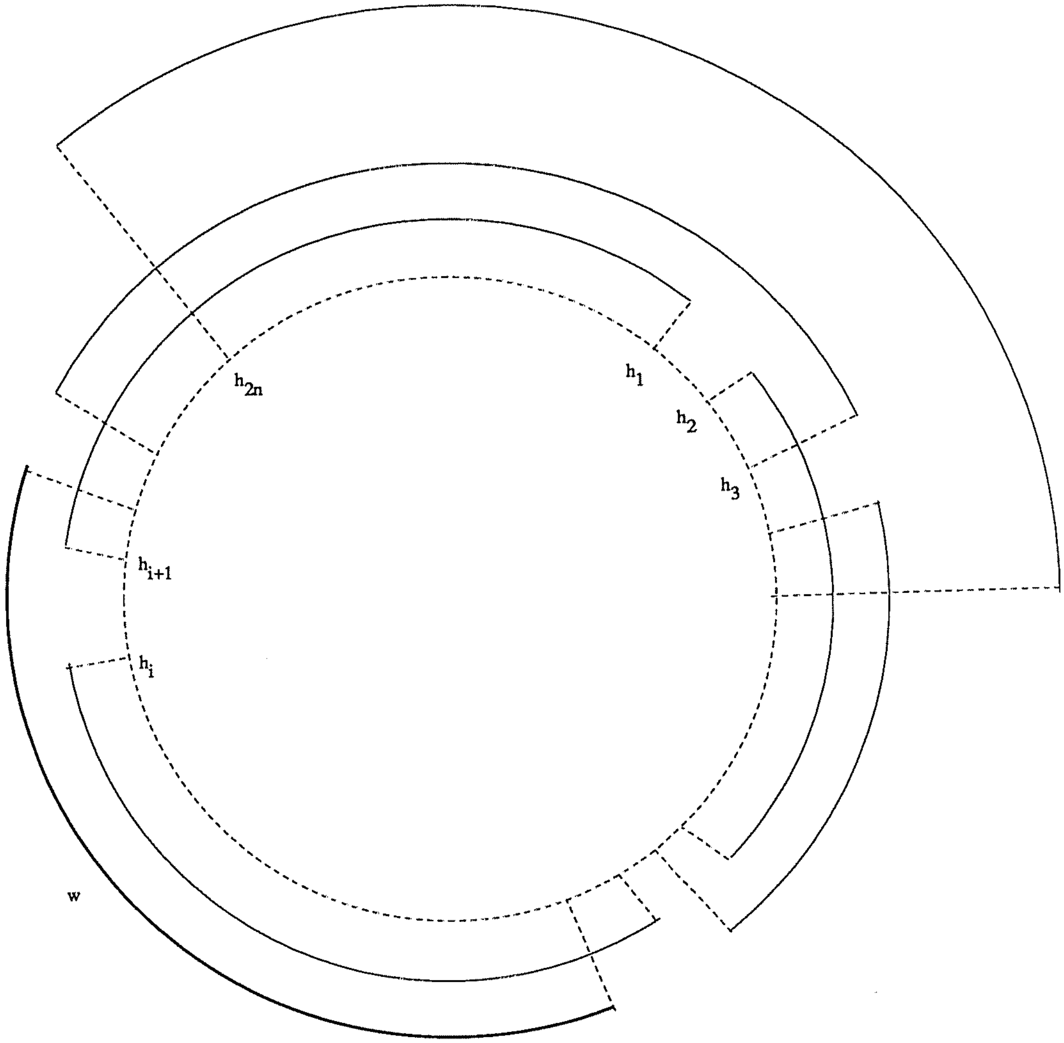
When the circular interval w of an elementary circular-arc graph $[G(J), w]$ is removed from J, we are left with a set of intervals which correspond to an interval graph. This implies that within interval w, there is a subinterval $y = (y_1, y_2)$ such that no interval from $J = I - \{w\}$ intersects y. Hence all intervals in J lie in the interval (y_2, y_1). Without loss of generality, we assume that (y_2, y_1) lies on a straight line and we apply our usual notions



Circular-Arc Graph

$a_i = (h_i, h_{i+1}) ;\quad W_i = \{ w, x, y \}$

FIGURE 1



Elementary Circular-Arc Graph
With Circular Interval w

FIGURE 2

of 'left' and 'right' to the intervals in J. When we make an 'ordered scan' of the intervals we mean a clockwise scan of the right endpoints of the intervals starting from y_2.

Algorithm $\mathcal{B}$

INPUT: A set of intervals, I and an interval w in I such that $[G(I), w]$ is an elementary circular-arc graph.

OUTPUT: A set of intervals $S \subseteq I$ with the property that S is the largest subset such that $G(S)$ is bipartite and $w \in S$.

1. Sort the intervals in $J = I - \{w\}$ according to the order in which the right endpoints appear in an ordered scan.
2. Let u_0 be the interval in J with the leftmost right endpoint.
3. $S \longleftarrow \{w, u_0\}$.
4. Remove w and u_0 from I and also all the intervals that form triangles with them.
5. **while** I is nonempty **do**
    6. Let v be the interval in I with the leftmost right endpoint.
    7. Put v in S and remove it from I.
    8. Remove from I all intervals that form triangles with v and one other interval from S.

   endwhile
9. Return S as the solution.

In step 8, we remove all intervals in I which form triangles with v and one other interval $y \in S$. In fact, we only need to consider $y \in \{w, u\}$, where w is the circular interval and u is the interval added to S in the previous iteration. The algorithm runs in time $O(n \log n)$. Note that if the input is already sorted according to the right endpoints of the intervals then the algorithm will run in time $O(n)$.

Lemma 5: Let I be the set of circular intervals corresponding to an elementary circular-arc graph $[G(I), w]$. Let u_0 be the interval in $J = I - \{w\}$ whose right endpoint is encountered first. Then among the solutions which include w, there exists a solution S' which includes u_0.

Proof: Let S be any solution for input I which includes w. Suppose $u_0 \notin S$. Denote the two bipartitions of $G(S)$ by V_1 and V_2 and denote their corresponding interval sets by J_1 and J_2. Without loss of generality we assume that $w \in J_1$. Let v be the interval in J_2 with the leftmost right endpoint. Let $S' = S - \{v\} \cup \{u_0\}$. If $x \in J_2 - \{v\}$ then x does not intersect v and hence x lies completely to the right of v. Consequently, it lies completely to the right of u_0. This implies that $G(S')$ is bipartite and hence S' is a solution which includes both w and u_0. ∎

Theorem 6: Algorithm $\mathcal{B}$ is correct.

Proof: We give only a rough sketch of the proof since it is similar to the proof of correctness of Algorithm $\mathcal{A}$ (Theorem 3). We use induction on the number of intervals in input J. Let S' be the solution returned by the algorithm and let S be any solution which includes the circular interval w and the interval with the leftmost right endpoint, u_0. Let Δ be the set of intervals in J which form triangles in $G(J)$ with w and u_0.

Following the lines of the proof of Theorem 3, we can show that $S' - \{u_0\}$ and $S - \{u_0\}$ are both solutions on input $J - \Delta - \{u_0\}$ and hence must be of the same size. This implies that S and S' are also of same size, thus proving the correctness of Algorithm $\mathcal{B}$. ∎

Algorithm $\mathcal{C}$

INPUT: A set of intervals, M, lying on a circle.

OUTPUT: $S \subseteq M$ such that the subgraph $G(S)$ is the largest induced bipartite subgraph in the circular-arc graph, $G(M)$.

1. **For** i, $1 \leq i \leq 2n$ **do**
   2. Run Algorithm $\mathcal{A}$ on M_i to obtain solution S_{i0}.
   3. Let $W_i = \{w_{i1}, \ldots, w_{ik}\}$.
   4. **For Each** $w_{ij} \in W_i$ run Algorithm $\mathcal{B}$ on the elementary circular-arc graph $[G(M_i \cup \{w_{ij}\}), w_{ij}]$. Denote its output by S_{ij}.
   5. Pick S_i to be the largest of the solutions $S_{i0}, S_{i1}, \ldots, S_{ik}$.

   endfor
6. Return S, the largest of the solutions $S_1, \ldots, S_{2n}$.

Theorem 7: Algorithm $\mathcal{C}$ is correct.

Proof: As noted earlier, for every solution, S, to the *MIBS* problem for a circular-arc graph, there exists an m, $1 \leq m \leq 2n$ such that S has at most one interval from W_m.

If there exists a solution S and an integer m, $1 \leq m \leq 2n$ such that S contains no interval from W_m, then S is also a solution to $G(M_m)$. Hence, Algorithm $\mathcal{C}$ will find a solution of size $|S|$ after executing step 2 in the m^{th} iteration. Otherwise, $\forall m$, $1 \leq m \leq 2n$ and for every solution S, $|S \cap W_m| \geq 1$. It follows that there exists an m, $1 \leq m \leq 2n$ such that $W_m \cap S = \{w_{mj}\}$, for some $w_{mj} \in W_m$. This implies that S is also a solution for $M_{mj} = M_m \cup \{w_{mj}\}$. Since M_m corresponds to an interval graph, the intervals in M_{mj} correspond to an elementary circular-arc graph with w_{mj} as its circular interval. Hence, Algorithm $\mathcal{C}$ would correctly find a solution of size $|S|$ after the execution of step 4 during its m^{th} iteration. ■

Each of W_i can be of size $O(n)$. Hence, there are $O(n)$ calls to Algorithm $\mathcal{A}$ and $O(n^2)$ calls to Algorithm $\mathcal{B}$. Since we need to sort only once, the total time complexity is $O(n^3)$.

We conclude with an observation about the existence of a "good" approximation algorithm for the *MIBS* problem for general graphs. The **Absolute Performance Ratio** of an approximation algorithm A for a maximization problem P is the infimum of the ratio between the size of the optimal solution and the size of the approximate solution over all instances of P. It is denoted by $R_A(P)$. Theorem 8 below provides a negative result regarding the existence of a "good" approximation algorithm for the Maximum Independent Set problem (*MIS*).

Theorem 8 [GJ79]: If $R_A(MIS) < k$ for some $k > 1$, then for any $k > 1$, there is an approximation algorithm C_k with $R_{C_k}(MIS) < k$.

It has been conjectured that there does not exist an algorithm A for which $R_A(MIS) < k$ for some $k > 1$. Using a reduction from the *MIS* problem it is easy to see that a similar result holds for the *MIBS* problem. More precisely, if $R_B(MIBS) < k$ for some $k > 1$, then for any $k > 1$ there is an algorithm D_k with $R_{D_k}(MIBS) < k$. This leads us to conjecture that there does not exist an an algorithm A to solve the *MIBS* problem with $R_A(MIBS) < k$ for some integer $k > 1$.

5. REFERENCES

[BL76]

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

[GJ79]

M. R. Garey and D. S. Johnson, Computers And Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman And Company, New York, 1979.

[GJMP80]

M. R. Garey, D. S. Johnson, G. L. Miller and C. H. Papadimitriou, The complexity of coloring circular-arcs and chords, *SIAM J. Alg. and Disc. Methods*, **1** (1980), 216-227.

[Ga74]

F. Gavril, Algorithms on Circular-Arc Graphs, *Networks*, **4** (1974), 357-369.

[Go80]

M. C. Golumbic, "Algorithmic Graph Theory And Perfect Graphs", Academic Press, New York, 1980.

[GLL82]

U. I. Gupta, D. T. Lee and Y. -T. Leung, Efficient algorithms for interval graphs and circular-arc graphs, *Networks*, **12** (1982), 459-467.

[LY80]

J. M. Lewis and M. Yannakakis, The node-deletion problem for hereditary properties is NP-Complete, *J. of Comp. System Sci.*, **20** (1980).

[Ma45]

E. Marczewski, Sur seux propriétés des classes d'ensembles, *Fund. Math.*, **33** (1945), 303-307.

[Tuc80]

A. Tucker, An efficient test for circular-arc graphs, *SIAM J. Comput.*, **9** (1980), 1-24.