TR-86-68

An Optimal Algorithm for Finding a Maximum Independent Set of a Circular-Arc Graph

by

Sumio Masuda & Kazuo Nakajima

TR-86-68

An Optimal Algorithm for Finding a Maximum Independent Set of a Circular-Arc Graph*

by

Sumio MASUDA

Electrical Engineering Department and Systems Research Center
University of Maryland
College Park, Maryland 20742
on leave from
Department of Information and Computer Sciences
Osaka University
Toyonaka, Osaka 560, Japan

and

Kazuo NAKAJIMA

Electrical Engineering Department,
Institute for Advanced Computer Studies,
and Systems Research Center
University of Maryland
College Park, Maryland 20742

* This work was supported in part by National Science Foundation Grants DCI-84-51510 and OIR-85-00108, and in part by a grant from AT&T Information Systems.

An Optimal Algorithm for Finding a Maximum Independent Set of a Circular-Arc Graph

by

Sumio Masuda and Kazuo Nakajima

Abstract

A new algorithm is presented for finding a maximum independent set of a circular-arc graph. When the graph is given in the form of a family of n arcs, our algorithm requires only $O(n \cdot \log n)$ time and $O(n)$ space. Furthermore, if the endpoints of the arcs are already sorted, it runs in $O(n)$ time. This algorithm is time- and space-optimal to within a constant factor.

1. Introduction

Let $G=(V,E)$ be a graph. Two distinct vertices u and v in V are said to be *independent* from each other if $(u,v) \notin E$; otherwise they are said to be *adjacent* to each other. A subset X of V is called an *independent set* of G if any two vertices in X are independent. A *maximum independent set* of G is an independent set whose cardinality is the largest among all independent sets of G.

Consider a finite family S of non-empty sets. A graph $G=(V,E)$ is called an *intersection graph for S* if there is a one-to-one correspondence between S and V such that two sets in S have a non-empty intersection if and only if the corresponding vertices in V are adjacent to each other. If S is a family of intervals on the real line, then G is called an *interval graph.* When S is a family of arcs on a circle, G is called a *circular-arc graph* for S.

Interval graphs have been used in many practical applications [10,12-13], and, as such, a wide variety of algorithms have been developed [2,5-6,8-9]. Furthermore, as a generalization of interval graphs, circular-arc graphs have received considerable attention in recent years. Tucker [11] proposed an $O(n^3)$ time algorithm for recognizing circular-arc graphs, where n is the number of the vertices in a given graph. Garey, Johnson, Miller and Papadimitriou [3] showed that the coloring problem is NP-complete for circular-arc graphs. Gavril [4] developed polynomial time algorithms for finding a maximum clique, a maximum independent set, and a minimum covering by disjoint cliques of a circular-arc graph. When the graph is given in the form of a family of n arcs, the algorithms produce solutions in $O(n^{3.5})$, $O(n^4)$, and $O(n^5)$ time, respectively. Later, Gupta, Lee and Leung [6] gave $O(n^2)$ time implementations of the last two algorithms of Gavril's. Recently, Hsu [7] presented an algorithm for finding a maximum weight clique for the case when each vertex is assigned a real number as its weight. Its time complexity is $O(n \cdot m)$, where m is the number of the edges of the graph.

In this paper, we present a new algorithm for finding a maximum independent set of a circular-arc graph. We show an $O(n \cdot \log n)$ time and $O(n)$ space implementation of the algorithm when the graph is given in the form of a family of n arcs on a circle. If the endpoints of the arcs are already sorted, the algorithm is shown to run in $O(n)$ time.

It should be noted that Gupta et al. [6] have proven that it requires $\Omega(n \cdot \log n)$ time in the worst case to find a maximum independent set of an interval graph with n vertices. Since every circular-arc graph is an interval graph, our algorithm is both time- and space-optimal to within a constant factor.

2. Definitions and Notations

Let $S=\{a_1,a_2, \ldots, a_n\}$ be a family of arcs on a circle C. Each endpoint of the arcs is assigned a positive integer, called a *coordinate*. The endpoints are located on the circumference of C in the ascending order of the values of the coordinates in the clockwise direction. Without loss of generality, we can assume that (i) all endpoints of the arcs in S are distinct, and (ii) no single arc in S covers the entire circle C by itself.

For simplicity, we call the endpoint with coordinate j as point j. Suppose that an arc begins at point j and ends at point k in the clockwise direction. Then, we denote such an arc by (j,k), and call points j and k as the *head* and the *tail*, respectively, of the arc (j,k). For $i=1,2,\ldots,n$, let h_i and t_i denote the coordinates of the head and tail, respectively, of arc a_i, that is, $a_i=(h_i,t_i)$. We show an example of a family of arcs in Fig. 1, where $a_1=(1,7)$, $a_2=(3,5)$, $a_3=(6,9)$, $a_4=(8,12)$, $a_5=(10,13)$, $a_6=(11,15)$, $a_7=(14,4)$ and $a_8=(16,2)$.

For an arc $a_i \in S$ and an endpoint j of another arc in S, we say that a_i *contains* point j if one of the following three conditions holds (see Fig. 2).

(i) $1 \leq h_i < j < t_i \leq 2n$.

(ii) $1 \leq t_i < h_i < j \leq 2n$.

(iii) $1 \leq j < t_i < h_i \leq 2n$.

For two distinct arcs a_i and a_j in S, we say that they *intersect* with each other if one of them contains at least one of the endpoints of the other arc; otherwise a_i and a_j are said to be *independent* from each other. If a_i contains both endpoints of a_j, we say that a_i *contains* a_j. The *circular-arc graph* for S, denoted by G_S, is defined as follows:

$$G_S \triangleq (V_S, E_S),$$

$$\text{where } V_S \triangleq \{v_1, v_2, \ldots, v_n\},$$

$$\text{and } E_S \triangleq \{ (v_i, v_j) \mid a_i \text{ and } a_j \text{ intersect with each other}\}.$$

For example, Fig. 3 depicts the circular-arc graph for the family of arcs given in Fig. 1.

A subfamily S' of S is called an *independent arc family* (abbreviated to an *IAF*) if any two arcs in S' are independent from each other. A *maximum independent arc family* (abbreviated to an *MIAF*) of S is an IAF whose cardinality is the largest among all IAF's of S. For example, the family of arcs shown in Fig. 1 has two MIAF's, $\{a_2, a_3, a_5, a_8\}$ and $\{a_2, a_3, a_6, a_8\}$. Clearly, the MIAF's of S and the maximum independent sets of G_S are in one-to-one correspondence. In the following section, we will present an algorithm for finding an MIAF of a family of arcs.

3. Outline of the Algorithm

Let $S = \{a_1, a_2, \ldots, a_n\}$ be a family of n arcs on a circle C. S is said to be *canonical* if (i) h_i's and t_i's for $i = 1,2,\ldots,n$ are all distinct integers between 1 and $2n$, and (ii) point 1 is the head of arc a_1. For instance, the family of arcs shown in Fig. 1 is canonical, but the one given in Fig. 4 is not. It should be noted, however, that these two families of arcs correspond to the same circular-arc graph, which is shown in Fig. 3. When S is not canonical, one can construct in $O(n \cdot \log n)$ time a family of arcs S' such that $G_S = G_{S'}$ by using a regular sorting algorithm [1]. Throughout this paper, we assume that the family S is canonical.

For a subfamily S' of S, let $\alpha(S')$ denote the cardinality of a maximum independent set of $G_{S'}$, or equivalently that of an MIAF of S'. We start with the following theorem.

Theorem 1. Suppose that an arc $a_i \in S$ contains another arc $a_j \in S$. Then, any MIAF of $S - \{a_i\}$ is an MIAF of S.

Proof. It is clear that $\alpha(S) \geq \alpha(S - \{a_i\})$. Let X be an MIAF of S. If $a_i \notin X$, then X is an IAF of $S - \{a_i\}$. On the other hand, if $a_i \in X$, then $(X - \{a_i\}) \cup \{a_j\}$ is an IAF of $S - \{a_i\}$. These imply that $\alpha(S) \leq \alpha(S - \{a_i\})$. Thus, $\alpha(S) = \alpha(S - \{a_i\})$, and hence any MIAF of $S - \{a_i\}$ is an MIAF of S. □

An arc $a_i = (h_i, t_i) \in S$ is called a *forward arc* if $h_i < t_i$; otherwise a_i is called a *backward arc*. For example, there are two backward arcs, a_7 and a_8, in the family of arcs of Fig. 1. In our algorithm, we first remove all forward arcs which contain other forward arcs. Let S_F denote the resultant family of the forward arcs, and let S_B denote the family of all backward arcs in S. Then, we have the following lemma and theorem.

Lemma 1. $S_F \neq \phi$.

Proof. Since S is canonical, it has at least one forward arc, that is, a_1. If a_1 is the only forward arc in S, $S_F = \{a_1\} \neq \phi$. On the other hand, if S has more than one forward arc, there exists at least one forward arc which does not contain any other forward arc.

Thus, $S_F \neq \phi$ in either case. □

Theorem 2. $1 \leq \alpha(S_F) \leq \alpha(S) \leq \alpha(S_F)+1$.

Proof. Since $S_F \neq \phi$ from Lemma 1 and $S_F \subseteq S$, $1 \leq \alpha(S_F) \leq \alpha(S)$. From Theorem 1, $\alpha(S)=\alpha(S_F \cup S_B)$, and hence $\alpha(S) \leq \alpha(S_F)+\alpha(S_B)$. Furthermore, it is clear that $\alpha(S_B)=1$ if $S_B \neq \phi$, and that $\alpha(S_B)=0$ if $S_B=\phi$. Therefore, the theorem holds. □

Our algorithm tests whether there exist an MIAF, X of S_F and an arc $a_j \in S_B$ such that $X \cup \{a_j\}$ is an IAF. From Theorem 2, if such an MIAF, X and an arc a_j exist, then $X \cup \{a_j\}$ is an MIAF of S; otherwise any MIAF of S_F is an MIAF of S. In general, the number of MIAF's of S_F may be an exponential function of $|S_F|$. However, our algorithm efficiently perform this test by exploiting the property of an MIAF of S which will be described later in Theorem 3.

Let $X=\{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$ with $h_{i_1} < t_{i_1} < h_{i_2} < t_{i_2} < \cdots < h_{i_k} < t_{i_k}$ be an IAF of S_F. Then we call h_{i_1} (resp., t_{i_k}) the *starting coordinate* (resp., the *ending coordinate*) of X and denote it by $sc(X)$ (resp., $ec(X)$). Let X_1 and X_2 be two IAF's of S_F. We say that X_1 *dominates* X_2 if one of the following conditions holds.

(i) $sc(X_1) > sc(X_2)$ and $ec(X_1) \leq ec(X_2)$.

(ii) $sc(X_1) \geq sc(X_2)$ and $ec(X_1) < ec(X_2)$.

An MIAF, X of S_F is called a *dominant maximum independent arc family* (abbreviated to a *DMIAF*) of S_F if no other MIAF of S_F dominates X.

Lemma 2. Let X be an MIAF of $S_F \cup S_B$. If there exists an MIAF, X_1 of S_F which dominates $X \cap S_F$, then $X_1 \cup (X \cap S_B)$ is an MIAF of S.

Proof. Suppose that $X \cap S_B=\phi$. Then $\alpha(S_F)=\alpha(S_F \cup S_B)=\alpha(S)$. Since $|X_1|=\alpha(S_F)$, $X_1 \cup (X \cap S_B)=X_1$ is an MIAF of S. On the other hand, suppose that $X \cap S_B \neq \phi$. Since $\alpha(S_B) \leq 1$, $|X \cap S_B|=1$. Let a_i be the unique element in $X \cap S_B$.

Then, it is clear that $t_i < sc(X - \{a_i\})$ and $ec(X - \{a_i\}) < h_i$. Since X_1 dominates $X \cap S_F = X - \{a_i\}$, $sc(X_1) \geq sc(X \cap S_F)$ and $ec(X_1) \leq ec(X \cap S_F)$. Therefore, we have $t_i < sc(X_1)$ and $ec(X_1) < h_i$ (see Fig. 5). This implies that $X_1 \cup \{a_i\}$ is an IAF of S. Since $|X_1 \cup \{a_i\}| = \alpha(S_F)+1$, $X_1 \cup (X \cap S_B) = X_1 \cup \{a_i\}$ is an MIAF of S from Theorem 2. □

Lemma 3. Let X be an MIAF of $S_F \cup S_B$. Then, there exists a DMIAF, X_1 of S_F such that $X_1 \cup (X \cap S_B)$ is an MIAF of S.

Proof. If $X \cap S_F$ is a DMIAF of S_F, then the theorem trivially holds. Otherwise, there exists a DMIAF of S_F, say X_2, which dominates $X \cap S_F$. From Lemma 2, $X_2 \cup (X \cap S_B)$ is an MIAF of S. □

Lemma 3 implies that it suffices to consider DMIAF's of S_F in order to test whether there exist an MIAF, X of S_F and an arc $a_j \in S_B$ such that $X \cup \{a_j\}$ is an IAF of S. In the following, we will show that the space which must be searched to find a desired MIAF of S_F can be reduced further.

The next lemma is obvious from the definition of a DMIAF.

Lemma 4. Let X_1 and X_2 be two DMIAF's of S_F. Then, $sc(X_1) = sc(X_2)$ if and only if $ec(X_1) = ec(X_2)$. Furthermore, $sc(X_1) < sc(X_2)$ if and only if $ec(X_1) < ec(X_2)$. □

Let D be the set of all DMIAF's of S_F. A subset of D, $R = \{X_1, X_2, \ldots, X_k\}$ is called an *essential DMIAF set* for S_F if it satisfies the following two conditions.

(i) $sc(X_i) \neq sc(X_j)$ for any $1 \leq i \neq j \leq k$.

(ii) For any DMIAF, X of S_F, there exists an integer j such that $1 \leq j \leq k$ and $sc(X_j) = sc(X)$.

Let us consider the family S of arcs shown in Fig. 6. In this particular case, S_F consists of eight arcs $a_1, a_2, \ldots, a_8$, and has ten MIAF's: $\{a_1, a_4, a_7\}$, $\{a_1, a_4, a_8\}$, $\{a_1, a_5, a_8\}$,

$\{a_1,a_6,a_8\}$, $\{a_2,a_4,a_7\}$, $\{a_2,a_4,a_8\}$, $\{a_2,a_5,a_8\}$, $\{a_2,a_6,a_8\}$, $\{a_3,a_5,a_8\}$ and $\{a_3,a_6,a_8\}$. Among them, $\{a_2,a_4,a_7\}$, $\{a_3,a_5,a_8\}$ and $\{a_3,a_6,a_8\}$ are DMIAF's. Since the last two have the same starting coordinate, each of the sets $\{ \{a_2,a_4,a_7\}, \{a_3,a_5,a_8\} \}$ and $\{ \{a_2,a_4,a_7\}, \{a_3,a_6,a_8\} \}$ is an essential DMIAF set for S_F.

From Lemma 3 and the first half of Lemma 4, we have the following theorem.

Theorem 3. Let X be an MIAF of $S_F \cup S_B$, and let R_F be an essential DMIAF set for S_F. Then, there exists a DMIAF, $X_1 \in R_F$ such that $X_1 \cup (X \cap S_B)$ is an MIAF of S. □

We now show the framework of our algorithm. Its correctness follows directly from Theorems 2 and 3.

Algorithm FIND-MIAF.

Input: A canonical family of arcs $S=\{a_1,a_2, \ldots, a_n\}$.

Output: An MIAF of S.

Method:

1. Determine S_F and S_B.
2. Find an essential DMIAF set for S_F, $R_F=\{X_1,X_2, \ldots, X_k\}$.
3. If there exist a DMIAF $X_i \in R_F$ and an arc $a_j \in S_B$ such that $t_j < sc(X_i)$ and $ec(X_i) < h_j$, then generate $X_i \cup \{a_j\}$; otherwise generate an arbitrary MIAF of S_F. □

In the next section, we give a linear time implementation of this algorithm.

4. Efficient Implementation of the Algorithm

4.1 Determination of S_F and S_B

Suppose that a canonical family of arcs $S=\{a_1,a_2, \ldots, a_n\}$ is given as an instance to Algorithm FIND-MIAF. Let S_F' denote the family of all forward arcs in S. Recall that S_F is the family of all forward arcs in S which do not contain any other forward arc in S. Note that $S = S_F' \cup S_B$ and $S_F' \cap S_B = \phi$. We can partition S into S_F' and S_B in $O(n)$ time. In order to extract S_F from S_F', we first create an empty queue Q and initialize S_F to be empty. Then we visit the endpoints of the arcs in S_F' one by one in the ascending order of their coordinates. If we find the head of some arc a_i, we insert integer i into Q. If we find the tail of some arc a_i, we delete from Q the integers which are placed before i. Then we delete i from Q and add it to S_F.

Suppose that an integer j is deleted when the tail of a_i $(i \neq j)$ is visited. Since the tail of a_j has not been visited, $t_i < t_j$. Furthermore, $h_j < h_i$ since the integers are inserted into Q in the ascending order of the coordinates of the heads of the corresponding arcs. These imply that a_j contains a_i. Thus, $a_j \notin S_F$. On the other hand, arc a_i does not contain any other forward arc; otherwise the tail of some arc a_k such that $h_k > h_i$ would have been visited earlier than that of a_i and integer i would have already been deleted. Therefore, $a_i \in S_F$.

Since S is canonical, the coordinates of the endpoints of the arcs in S_F' are distinct integers between 1 and $2n$. Therefore, this procedure determines S_F in $O(n)$ time and with $O(n)$ space. Thus, the next theorem follows.

Theorem 4. Determination of S_F and S_B can be done in $O(n)$ time and with $O(n)$ space. □

4.2 Finding an Essential DMIAF Set for S_F

Suppose that S_F has been obtained. Without loss of generality, we assume that $S_F = \{a_1, a_2, \ldots, a_{|S_F|}\}$ with $h_1 < h_2 < \cdots < h_{|S_F|}$. (The renumbering of the subscripts of the arcs can be performed in $O(n)$ time by a bucket sort [1], if necessary.) The following lemma provides an important property of an essential DMIAF set for S_F.

Lemma 5. Let $R_F=\{X_1,X_2, \ldots, X_k\}$ be an essential DMIAF set for S_F. Then, $X_i \cap X_j=\phi$ for $1 \le i \ne j \le k$.

Proof. Let i and j be two integers such that $1 \le i \ne j \le k$. By definition, X_i and X_j are DMIAF's of S_F. Without any loss of generality, we can assume that $sc(X_i)<sc(X_j)$ and $ec(X_i)<ec(X_j)$.

Assume that $X_i \cap X_j \ne \phi$, and let a_k be an element in $X_i \cap X_j$. Let X_i^- and X_i^+ denote $\{a_q \in X_i \mid t_q<h_k\}$ and $\{a_q \in X_i \mid h_q>t_k\}$, respectively. Similarly, let X_j^- and X_j^+ denote $\{a_q \in X_j \mid t_q<h_k\}$ and $\{a_q \in X_j \mid h_q>t_k\}$, respectively. Then, clearly $X_j^- \cup \{a_k\} \cup X_i^+$ is an IAF of S_F. This implies that $|X_j^+| \ge |X_i^+|$ since $X_j=X_j^- \cup \{a_k\} \cup X_j^+$ is an MIAF of S_F. Similarly, since $X_i^- \cup \{a_k\} \cup X_j^+$ is an IAF of S_F and X_i is an MIAF of S_F, we have $|X_j^+| \le |X_i^+|$. Therefore, $|X_j^+|=|X_i^+|$. By the same reasoning, we can show that $|X_j^-|=|X_i^-|$. These imply that $X_j^- \cup \{a_k\} \cup X_i^+$ is an MIAF of S_F since it is an IAF of S_F and $|X_j^- \cup \{a_k\} \cup X_i^+|=|X_j|=|X_i|$. It is clear that $sc(X_i)<sc(X_j)=sc(X_j^- \cup \{a_k\} \cup X_i^+)$ and $ec(X_i)=ec(X_j^- \cup \{a_k\} \cup X_i^+)<ec(X_j)$. Therefore, $X_j^- \cup \{a_k\} \cup X_i^+$ dominates both X_i and X_j. This contradicts the facts that X_i and X_j are DMIAF's of S_F. Consequently, $X_i \cap X_j=\phi$. □

Corollary 1. Let $R_F=\{X_1,X_2, \ldots, X_k\}$ be an essential DMIAF set for S_F. Then $|X_1|+|X_2|+\cdots+|X_k| \le |S_F|$.

Proof. It is clear from Lemma 5. □

Let Z be defined as $\{a_i \in S_F \mid h_1 \le h_i < t_1\}$. Then, the following lemma is obtained.

Lemma 6. For any MIAF, X of S_F, $|X \cap Z|=1$.

Proof. Any two arcs in Z intersect with each other, and hence $|X \cap Z| \leq 1$. Furthermore, $|X \cap Z| \neq 0$; otherwise, $X \cup \{a_1\}$ would be an IAF of S_F. Therefore, $|X \cap Z|=1$. □

For an IAF of S_F, $X=\{a_{i_1}, a_{i_2}, \ldots, a_{i_j}\}$ with $h_{i_1}<h_{i_2}<\cdots<h_{i_j}$, a_{i_1} is called the *starting arc* of X. For each arc $a_i \in S_F$, an IAF containing a_i as its starting arc is called a *largest IAF* for a_i if it contains the maximum number of arcs. Then we define YS_i as the set of all largest IAF's for a_i. For example, consider the family of arcs of Fig. 6 again. Then, $YS_1=\{\{a_1,a_4,a_7\}, \{a_1,a_4,a_8\}, \{a_1,a_5,a_8\}, \{a_1,a_6,a_8\}\}$ and $YS_3=\{\{a_3,a_5,a_8\}, \{a_3,a_6,a_8\}\}$.

For $i=1,2,\ldots,|S_F|$, let Y_i be an IAF in YS_i whose ending coordinate is the minimum among all IAF's in YS_i. Suppose that $Z=\{a_1,a_2,\ldots,a_m\}$. By assumption, $h_1<h_2<\cdots<h_m$. We show below several theorems and lemmas which play important roles in finding an essential DMIAF set for S_F.

Lemma 7. Let X be a DMIAF and let a_i be its starting arc. Then, $1 \leq i \leq m$ and Y_i is a DMIAF of S_F.

Proof. From Lemma 6, $a_i \in Z$, that is, $1 \leq i \leq m$. It is clear that X is a largest IAF for a_i. Therefore, $|Y_i|=|X|$, and hence Y_i is an MIAF of S_F. Furthermore, since Y_i has the minimum ending coordinate among all IAF's in YS_i and X is not dominated by Y_i, $ec(Y_i)=ec(X)$. These imply that Y_i is a DMIAF of S_F. □

Theorem 5. Let R be the set of all DMIAF's in $\{Y_1,Y_2,\ldots,Y_m\}$. Then, R is an essential DMIAF set for S_F.

Proof. Let X be any DMIAF of S_F. Then, from Lemma 7, a DMIAF of S_F, Y_i exists in R such that $1 \leq i \leq m$ and $sc(X)=sc(Y_i)$. Therefore, there exists a subset of R

which is an essential DMIAF set for S_F. Since $h_1 < h_2 < \cdots < h_m$, $sc(Y_1) < sc(Y_2) < \cdots < sc(Y_m)$. This implies that no two DMIAF's in R have the same starting coordinate. Thus, R itself is an essential DMIAF set for S_F. □

Lemma 8. Suppose that Y_i is not an MIAF of S_F for some integer i such that $1 \le i < m$. Then, Y_j is not an MIAF of S_F for $j = i+1, i+2, \ldots, m$.

Proof. Assume that Y_k is an MIAF of S_F for some k such that $i+1 \le k \le m$. Then $|Y_k| > |Y_i| \ge 1$, and hence $|Y_k| \ge 2$. Since $h_i < h_k$ and a_i does not contain a_k, $t_i < t_k$. Furthermore, it is clear that $t_k < sc(Y_k - \{a_k\})$. Therefore, $\{a_i\} \cup (Y_k - \{a_k\})$ is an MIAF of S_F. Since its starting arc is a_i, every IAF in YS_i is an MIAF of S_F. This contradicts the hypothesis that Y_i is not an MIAF of S_F. Therefore, there does not exist such an integer k that $i+1 \le k \le m$ and Y_k is an MIAF of S_F. □

Corollary 2. Y_1 is an MIAF of S_F.

Proof. It is clear from Lemmas 7 and 8. □

Lemma 9. Suppose that Y_i is an MIAF of S_F for some integer i such that $1 \le i \le m$. Then, if Y_i is not a DMIAF of S_F, there exists an integer j such that $i < j \le m$ and that Y_j is a DMIAF of S_F which dominates Y_i.

Proof. If Y_i is not a DMIAF of S_F, then there exists a DMIAF of S_F, say X, which dominates Y_i. By definition, Y_i has the smallest ending coordinate among all longest IAF's for a_i, and hence $a_i \notin X$. This implies that $h_i = sc(Y_i) < sc(X)$ and $ec(X) \le ec(Y_i)$. Let a_j be the starting arc of X. Then, from Lemma 7, $j \le m$ and Y_j is a DMIAF of S_F. Since $sc(X) = h_j$, $h_i < h_j$, and hence $i < j \le m$. Furthermore, since $sc(Y_j) = sc(X)$, $ec(Y_j) = ec(X)$ from Lemma 4. Therefore, Y_j dominates Y_i. □

Corollary 3. If Y_m is an MIAF of S_F, then it is a DMIAF of S_F.

Proof. It is clear from Lemma 9. □

Corollary 4. Suppose that Y_i is an MIAF of S_F for some integer i such that $1 \leq i < m$. Then, if Y_{i+1} is not an MIAF of S_F, Y_i is a DMIAF of S_F.

Proof. It is clear from Lemmas 8 and 9. □

Lemma 10. Suppose that both Y_i and Y_{i+1} are MIAF's of S_F for some integer i such that $1 \leq i < m$. Then, Y_i is a DMIAF of S_F if and only if $ec(Y_i) < ec(Y_{i+1})$.

Proof. Clearly $h_i = sc(Y_i) < sc(Y_{i+1}) = h_{i+1}$. So, if $ec(Y_i) \geq ec(Y_{i+1})$, then Y_{i+1} dominates Y_i. Thus, if Y_i is a DMIAF of S_F, then $ec(Y_i) < ec(Y_{i+1})$.

Suppose that Y_i is not a DMIAF of S_F. From Lemma 9, there exists an integer j such that $i+1 \leq j \leq m$ and that Y_j is a DMIAF of S_F which dominates Y_i. Since a_i does not contain a_j and Y_j dominates Y_i, $|Y_i| \geq 2$, and hence $|Y_j| \geq 2$. Furthermore, if $j \neq i+1$, $t_{i+1} < t_j$, and hence $t_{i+1} < sc(Y_j - \{a_j\})$. This inequality also holds when $j = i+1$. Therefore, $ec(Y_{i+1}) \leq ec(Y_j - \{a_j\})$; otherwise $\{a_{i+1}\} \cup (Y_j - \{a_j\})$ would be selected as Y_{i+1}. Since $ec(Y_j - \{a_j\}) = ec(Y_j) \leq ec(Y_i)$, we have $ec(Y_{i+1}) \leq ec(Y_i)$. Thus, if $ec(Y_i) < ec(Y_{i+1})$, then Y_i is a DMIAF of S_F. □

Theorem 6. Suppose that Y_i is an MIAF of S_F for some integer i such that $1 \leq i \leq m$. Then, Y_i is a DMIAF of S_F if and only if one of the following three conditions holds.

(i) $i < m$ and Y_{i+1} is not an MIAF of S_F.

(ii) $i < m$ and $ec(Y_i) < ec(Y_{i+1})$.

(iii) $i = m$.

Proof. It is clear from Lemma 10 and Corollaries 3 and 4. □

We now present a procedure to find an essential DMIAF set for S_F. Its correctness can easily be proven by using Theorems 5 and 6, Lemma 8 and Corollary 2.

Procedure FIND-EDS.

1. $Z \leftarrow \{a_i \in S_F \mid h_1 \leq h_i < t_1\}$. Suppose that $Z = \{a_1, a_2, \ldots, a_m\}$.
2. For $i = 1,2,\ldots,m$, find $ec(Y_i)$ and $|Y_i|$, where Y_i is defined as before.
3. $R \leftarrow \phi$. $i \leftarrow 1$.
4. While $|Y_i| = |Y_1|$ and $i < m$, execute the following instructions (1) and (2).

   (1) If $|Y_{i+1}| < |Y_1|$ or $ec(Y_i) < ec(Y_{i+1})$, then determine Y_i and $R \leftarrow R \cup \{Y_i\}$.

   (2) $i \leftarrow i+1$.
5. If $i = m$ and $|Y_m| = |Y_1|$, then determine Y_m and $R \leftarrow R \cup \{Y_m\}$.
6. Generate R. □

As an example, consider the family of arcs of Fig. 6. Since $t_1 = 6$, Z is determined as $\{a_1, a_2, a_3\}$ at Step 1. Then, Step 2 finds $ec(Y_1) = 17$, $|Y_1| = 3$; $ec(Y_2) = 17$, $|Y_2| = 3$; and $ec(Y_3) = 19$, $|Y_3| = 3$. At Step 4, Y_1 is not added to R since $|Y_2| = |Y_1| = 3$ and $ec(Y_1) = ec(Y_2)$. On the other hand, Y_2 is added to R since $ec(Y_2) < ec(Y_3)$. Similarly, Y_3 is added to R at Step 5. While Y_2 is uniquely determined as $\{a_2, a_4, a_7\}$, Y_3 may be chosen from two candidates, $\{a_3, a_5, a_8\}$ and $\{a_3, a_6, a_8\}$. Therefore, the resultant essential DMIAF set is either $\{\{a_2, a_4, a_7\}, \{a_3, a_5, a_8\}\}$ or $\{\{a_2, a_4, a_7\}, \{a_3, a_6, a_8\}\}$.

In what follows, we describe an efficient implementation of Procedure FIND-EDS.

For each arc $a_i \in S_F$, let $NEXT(a_i)$ be defined as an arc a_k such that $h_k = Min\{h_j \mid a_j \in S_F \text{ and } h_j > t_i\}$ if $\{a_j \in S_F \mid h_j > t_i\} \neq \phi$, and otherwise defined as "null". For example, for the family of arcs of Fig. 6, $NEXT(a_1) = a_4$, $NEXT(a_2) = a_4$, $NEXT(a_3) = a_5$, $NEXT(a_4) = a_7$, $NEXT(a_5) = a_8$, $NEXT(a_6) = a_8$, $NEXT(a_7) =$ "null", and

$NEXT(a_8)$="null".

For each arc $a_i \in S_F$, let N_i be defined as $\{a_{i_1}=a_i, a_{i_2}, \ldots, a_{i_k}\}$ such that $NEXT(a_{i_j})=a_{i_{j+1}}$ for $j=1,2,\ldots,k-1$ and $NEXT(a_{i_k})$="null". Then we have the following lemma.

Lemma 11. For each arc $a_i \in Z$, N_i is an IAF of S_F. Furthermore, $|Y_i|=|N_i|$ and $ec(Y_i)=ec(N_i)$.

Proof. It is clear from the definitions that N_i is an IAF of S_F. Suppose that $N_i=\{a_{i_1}=a_i, a_{i_2}, \ldots, a_{i_k}\}$ with $h_{i_1}<h_{i_2}<\cdots<h_{i_k}$ and $Y_i=\{a_{i'_1}=a_i, a_{i'_2}, \ldots, a_{i'_j}\}$ with $h_{i'_1}<h_{i'_2}<\cdots<h_{i'_j}$. From the definition of Y_i, $j \geq k$. Since no arc in S_F contains any other arc in S_F, we can show that $h_{i_p} \leq h_{i'_p}$ and $t_{i_p} \leq t_{i'_p}$ for $p=1,2,\ldots,k$ by an easy induction proof on the value of p. Furthermore, since $t_{i_k} \leq t_{i'_k}$ and $NEXT(a_{i_k})$="null", $NEXT(a_{i'_k})$="null". This implies that $k=j$, that is, $|N_i|=|Y_i|$. This further implies from the definition of Y_i that $ec(N_i) \geq ec(Y_i)$, that is, $t_{i_k} \geq t_{i'_k}$. Therefore, $ec(N_i)=ec(Y_i)$. □

For each arc $a_i \in S_F$, $NEXT(a_i)$ can be determined as follows. We first create an empty set P, and then start visiting the endpoints of the arcs of S_F in the ascending order of their coordinates. Suppose we find the head of some arc a_j. If P is empty, we do nothing. On the other hand, if P is not empty, we set $NEXT(a_k)$ to a_j for each element a_k in P and then delete all such elements from P. If we find the tail of some arc a_j which is not the last endpoint, we add a_j to P. If the tail is the last endpoint, we set $NEXT(a_k)$ to "null" for each element a_k in $P \cup \{a_j\}$. Since the coordinates of the endpoints of the arcs in S_F are distinct integers between 1 and $2n$, this procedure requires $O(n)$ time and space.

We now define a digraph H_F as follows:

$H_F = (W_F, A_F)$,

where $W_F = \{w_i \mid a_i \in S_F\}$,

and $A_F = \{(w_i \rightarrow w_j) \mid NEXT(a_i) = a_j\}$.

As an example, Fig. 7 illustrates the graph H_F which corresponds to the family of arcs of Fig. 6.

Since the outdegree of each vertex in H_F is at most one, $|A_F| < |S_F|$. Thus, we have the following lemma.

Lemma 12. The construction of H_F with the aforementioned computation of $NEXT(\cdot)$ requires $O(n)$ time and space. □

The next lemma is obvious from the definition of H_F.

Lemma 13. H_F has the following properties.

(i) For each vertex w_i with outdegree 0, $|N_i| = 1$ and $ec(N_i) = t_i$.

(ii) For each edge $(w_i \rightarrow w_j)$ in A_F, $|N_i| = |N_j| + 1$ and $ec(N_i) = ec(N_j)$.

(iii) For each arc $a_i \in Z$, the maximal directed path in H_F starting from w_i corresponds to N_i. □

According to Lemma 13 (i) and (ii), one can easily determine $|N_i|$ and $ec(N_i)$ for all arcs $a_i \in S_F$ in $O(|S_F|)$ time. From Lemma 11, for each arc $a_i \in Z$, $|Y_i| = |N_i|$ and $ec(Y_i) = ec(N_i)$. Therefore, the following lemma is obtained from Lemma 12.

Lemma 14. Step 2 of Procedure FIND-EDS requires $O(n)$ time and space. □

Steps 4 and 5 of Procedure FIND-EDS can be performed based on Lemma 11 and Lemma 13 (iii). Each time we find an integer i for which the conditions of Step 4 (i) or Step 5 are satisfied, we can determine Y_i by finding a maximal directed path in H_F

starting from w_i. Therefore, Steps 4 and 5 can be executed in $O(|Z| + \sum_{Y_i \in R} |Y_i|)$ time. Since $\sum_{Y_i \in R} |Y_i| \leq |S_F|$ from Corollary 1, the following theorem is obtained from Lemma 14.

Theorem 7. An essential DMIAF set for S_F can be formed in $O(n)$ time and with $O(n)$ space. □

4.3 Determination of an MIAF of S

Suppose that an essential DMIAF set for S_F, $R_F = \{X_1, X_2, \ldots, X_k\}$ has been obtained. For each arc $a_i \in S_B$, let $NEXT(a_i)$ be defined as X_j such that $sc(X_j) = Min\{sc(X_p) \mid X_p \in R_F \text{ and } sc(X_p) > t_i\}$ if $\{X_p \in R_F \mid sc(X_p) > t_i\} \neq \phi$, and otherwise defined as "null". Then, the following theorem holds.

Theorem 8. Suppose that there exist an arc $a_j \in S_B$ and a DMIAF, $X_i \in R_F$ such that $t_j < sc(X_i)$ and $ec(X_i) < h_j$. Then, $t_j < sc(NEXT(a_j))$ and $ec(NEXT(a_j)) < h_j$.

Proof. Assume that $X_i \neq NEXT(a_j)$. From the definition of $NEXT(a_j)$, $t_j < sc(NEXT(a_j)) < sc(X_i)$. Therefore, $ec(NEXT(a_j)) < ec(X_i)$ from Lemma 4, and hence $ec(NEXT(a_j)) < h_j$. □

By a procedure similar to the one for computing $NEXT(\cdot)$ for the arcs in S_F, one can determine $NEXT(a_i)$ for all arcs $a_i \in S_B$ in $O(n)$ time. In this case, after the creation of an empty set P, we visit the tails of the arcs in S_B and the heads of the starting arcs of the DMIAF's in R_F in the ascending order of their coordinates until the last head is visited. The other part of the procedure is almost the same as that of the previous one.

After finding $NEXT(\cdot)$ for all arcs in S_B, according to Theorem 8, we can easily find in $O(|S_B|)$ time an arc $a_j \in S_B$ and a DMIAF, $X_i \in R_F$ such that $t_j < sc(X_i)$ and $ec(X_i) < h_j$ if they exist. If such an arc and a DMIAF do not exist, any DMIAF in R_F is an MIAF of S. Thus, we have the following theorem.

Theorem 9. Suppose that an essential DMIAF set for S_F has been obtained. Then, an MIAF of S can be obtained in $O(n)$ time and with $O(n)$ space. □

4.4 Time and space complexities of the algorithm

We now show the following two theorems.

Theorem 10. The time and space complexities of Algorithm FIND-MIAF each are $O(n)$.

Proof. It is clear from Theorems 4, 7 and 9. □

Theorem 11. Given a family S of n arcs on a circle, a maximum independent set of its corresponding circular-arc graph G_S can be found in $O(n \cdot \log n)$ time and with $O(n)$ space. These complexities are optimal to within a constant factor.

Proof. As mentioned before, one can construct a canonical family of arcs S' such that $G_S = G_{S'}$ in $O(n \cdot \log n)$ time and with $O(n)$ space. The application of Algorithm FIND-MIAF to the resultant family of arcs S' requires $O(n)$ time and space due to Theorem 10. Therefore, we can find a maximum independent set of G_S in $O(n \cdot \log n)$ time and with $O(n)$ space.

Every circular-arc graph is an interval graph. Furthermore, it is known that it requires $\Omega(n \cdot \log n)$ time in the worst case to find a maximum independent set of an interval graph when the graph is given in the form of a family of n intervals [6]. Therefore, our algorithm is time- and space-optimal to within a constant factor. □

5. Conclusion

In this paper, we have presented an optimal algorithm for finding a maximum independent set of a circular-arc graph. When the graph is given in the form of a family of n arcs on a circle, our algorithm runs in $O(n \cdot \log n)$ time and with $O(n)$ space. Moreover, it requires only $O(n)$ time if the endpoints of the arcs are already sorted, in other words, if the order of their appearances on the circle is known.

It does not seem that our algorithm can be extended to the problem of finding a maximum weight independent set when each vertex is assigned a weight. It is interesting to develop an optimal algorithm for such a problem.

References

[1] A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] K. S. Booth and G. S. Lueker, "Testing for Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms", *J. Computer and System Sciences*, Vol. 13, pp. 335-379, 1976.

[3] M. R. Garey, D. S. Johnson, G. L. Miller and C. H. Papadimitriou, "The Complexity of Coloring Circular Arcs and Chords", *SIAM J. on Algebraic and Discrete Methods*, Vol. 1, pp. 216-227, 1980.

[4] F. Gavril, "Algorithms on Circular-Arc Graphs", *Networks*, Vol. 4, pp. 357-369, 1974.

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

[6] U. I. Gupta, D. T. Lee and J. Y.-T. Leung, "Efficient Algorithms for Interval Graphs and Circular-Arc Graphs", *Networks*, Vol. 12, pp. 459-467, 1982.

[7] W.-L. Hsu, "Maximum Weight Clique Algorithms for Circular-Arc Graphs and Circle Graphs", *SIAM J. on Computing*, Vol. 14, pp. 224-231, 1985.

[8] M. Keil, "Finding Hamiltonian Circuits in Interval Graphs", *Information Processing Letters*, Vol. 20, pp. 201-206, 1985.

[9] G. S. Lueker and K. S. Booth, "A Linear Time Algorithm for Deciding Interval Graph Isomorphism", *J. Association for Computing Machinery*, Vol. 26, pp. 183-195, 1979.

[10] T. Ohtsuki, H. Mori, E. S. Kuh, T. Kashiwabara and T. Fujisawa, "One Dimensional Logic Gate Assignment and Interval Graphs", *IEEE Trans. on Circuits and Systems*, Vol. CAS-26, pp. 675-684, 1979.

[11] A. Tucker, "An Efficient Test for Circular-Arc Graphs", *SIAM J. on Computing*, Vol. 9, pp. 1-24, 1980.

[12] O. Wing, S. Huang and R. Wang, "Gate Matrix Layout", *IEEE Trans. on Computer-Aided Design*, Vol. CAD-4, pp. 220-231, 1985.

[13] O. J. Yu and O. Wing, "Interval-Graph-Based PLA Folding", *Proc. 1985 IEEE Int'l Symp. on Circuits and Systems*, Kyoto, Japan, 1985, pp. 1463-1466.

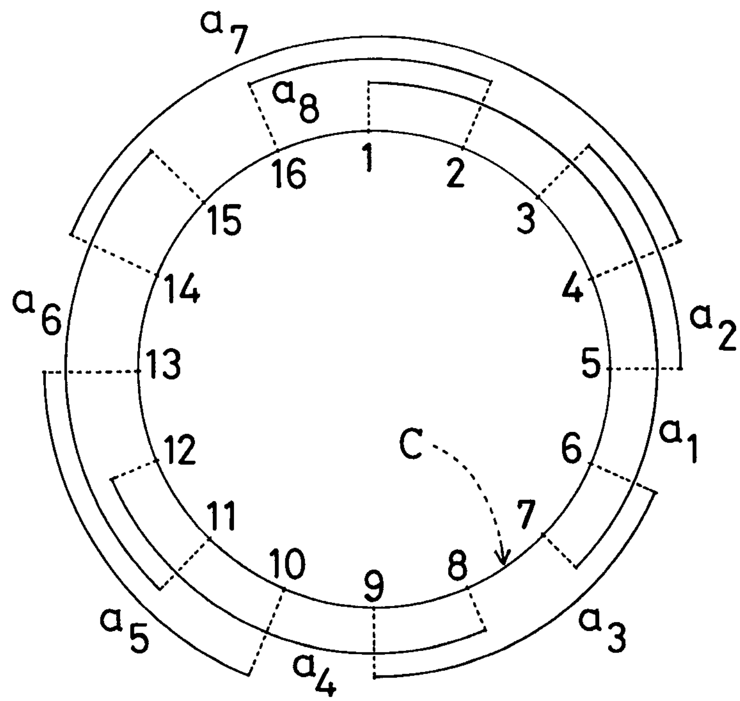

Fig. 1. A family of arcs on a circle C.

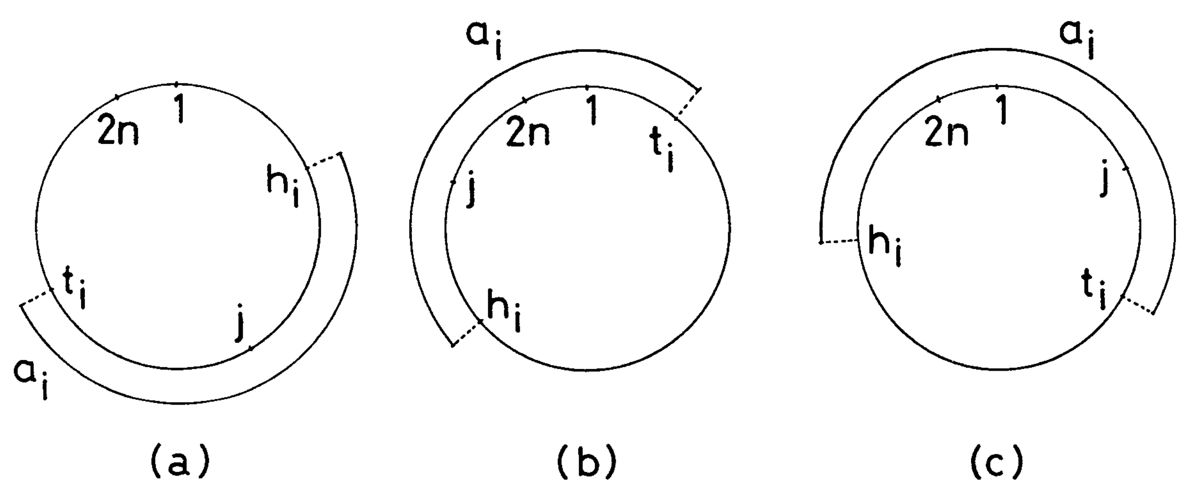

Fig. 2. Cases in which arc a_i contains point j.
(a) $1 \leq h_i < j < t_i \leq 2n$. (b) $1 \leq t_i < h_i < j \leq 2n$.
(c) $1 \leq j < t_i < h_i \leq 2n$.

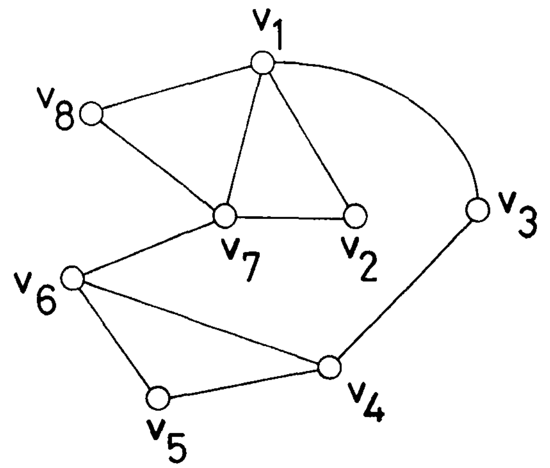

Fig. 3. The circular-arc graph for the family of arcs in Fig. 1.

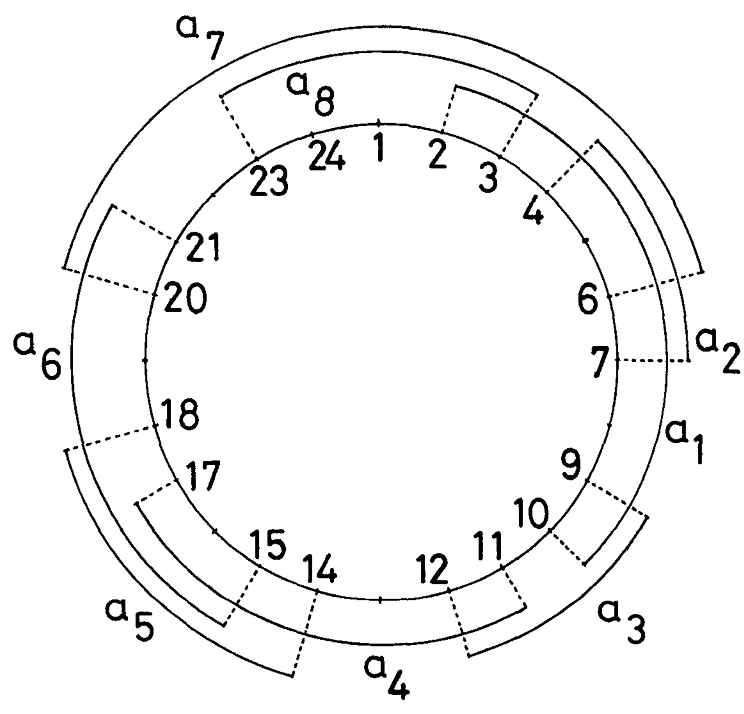

Fig. 4. A non-canonical family of arcs.

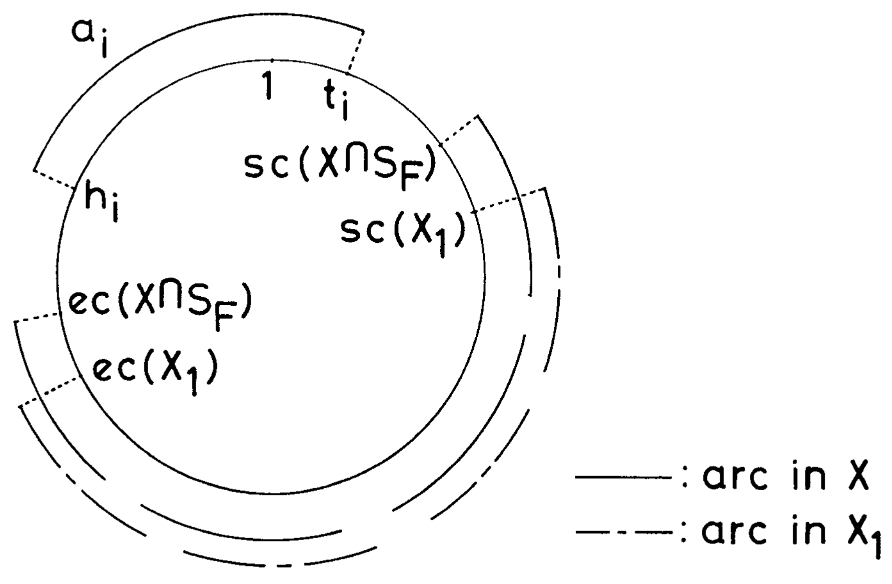

Fig. 5. An illustration for the proof of Lemma 2.

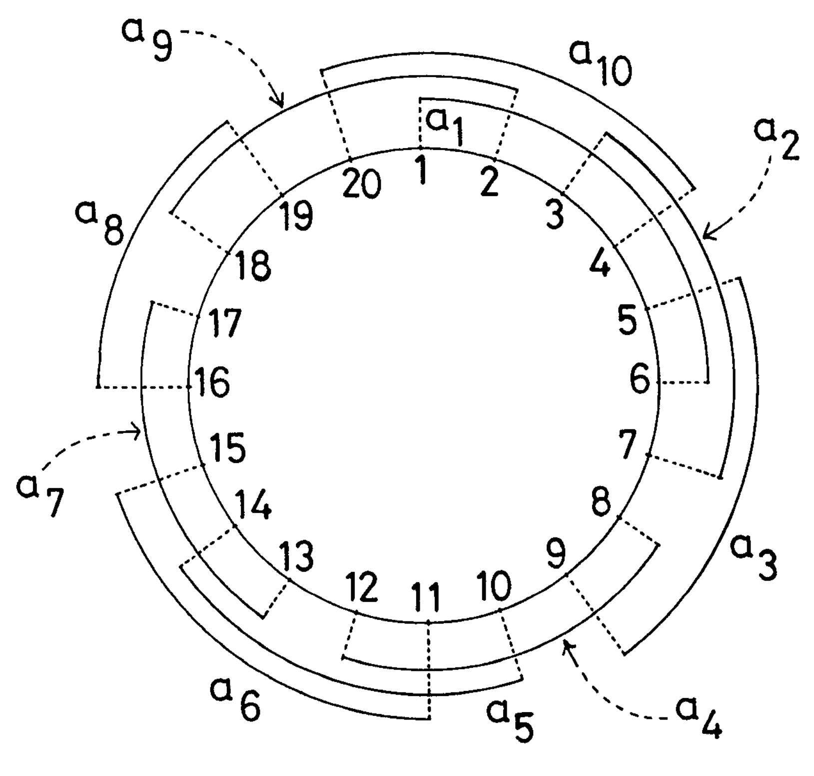

Fig. 6. A canonical family of arcs S. $\{\ \{a_2,a_4,a_7\},\ \{a_3,a_5,a_8\}\ \}$ and $\{\ \{a_2,a_4,a_7\},\ \{a_3,a_6,a_8\}\ \}$ are essential DMIAF sets for S_F.

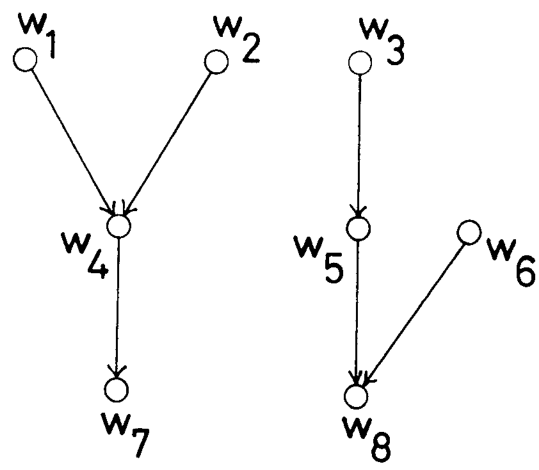

Fig. 7. The graph H_F for the family of arcs of Fig. 6.