Journal of Computer and Communications, 2016, 4, 23-31
Published Online June 2016 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2016.48003

An Algorithm for the Feedback Vertex Set Problem on a Normal Helly Circular-Arc Graph

Hirotoshi Honma, Yoko Nakajima, Atsushi Sasaki

Department of Creative Engineering, National Institute of Technology, Kushiro College, Kushiro, Japan
Email: honma@kushiro-ct.ac.jp, yoko@kushiro-ct.ac.jp, sasaki@kushiro-ct.ac.jp

Received 25 April 2016; accepted 24 June 2016; published 27 June 2016

Abstract

The feedback vertex set (FVS) problem is to find the set of vertices of minimum cardinality whose removal renders the graph acyclic. The FVS problem has applications in several areas such as combinatorial circuit design, synchronous systems, computer systems, and very-large-scale integration (VLSI) circuits. The FVS problem is known to be NP-hard for simple graphs, but polynomial-time algorithms have been found for special classes of graphs. The intersection graph of a collection of arcs on a circle is called a circular-arc graph. A normal Helly circular-arc graph is a proper subclass of the set of circular-arc graphs. In this paper, we present an algorithm that takes $O(n+m)$ time to solve the FVS problem in a normal Helly circular-arc graph with n vertices and m edges.

Keywords

Design and Analysis of Algorithms, Feedback Vertex Set, Normal Helly Circular-Arc Graphs, Intersection Graphs

1. Introduction

Let $\mathcal{F}$ be a family of nonempty sets. A simple graph G is the *intersection graph* of $\mathcal{F}$ if there exists a one-to-one correspondence between the vertices of G and the sets in $\mathcal{F}$, such that two vertices in G are adjacent if and only if their corresponding sets have a nonempty intersection. If $\mathcal{F}$ is a family of intervals on the real line, G is called an *interval graph* [1]. Furthermore, a graph G is called a *circular-arc graph* if it is the in-

How to cite this paper: Honma, H., Nakajima, Y. and Sasaki, A. (2016) An Algorithm for the Feedback Vertex Set Problem on a Normal Helly Circular-Arc Graph. *Journal of Computer and Communications*, **4**, 23-31.
http://dx.doi.org/10.4236/jcc.2016.48003

tersection graph of a collection of arcs on a circle [1]. Circular-arc graphs properly contain a class of interval graphs as a subclass. Circular-arc graphs have applications in areas such as genetics [2], traffic control [1], multidimensional scaling [3], compiler design [4], ring network modeling [5]. In recent years, circular-arc graphs have been investigated extensively from both theoretical and algorithmic perspectives [6]-[9].

Let $G=(V,E)$ be a simple graph, where V is the set of vertices and E is the set of edges of G, with $|V|=n$ and $|E|=m$. Suppose that V' is a nonempty subset of V. The subgraph of G whose vertex set is V' and whose edge set is the set of those edges of G that have both vertices in V' is called the *induced subgraph* on V' and is denoted by $G[V']$ [10]. A cycle with no repeated vertices is a simple cycle. In this paper, the term "cycle" denotes "simple cycle". A *feedback vertex set* (FVS) consists of a subset $F \subseteq V$ such that each cycle in G contains at least one vertex in F. In other words, a subset $F \subseteq V$ is an FVS of G if the subgraph induced by $G[V-F]$ is acyclic. The FVS problem is to find an FVS of minimum cardinality (MFVS) in G. The FVS problem has applications in several areas such as deadlock prevention in operating systems [11], combinatorial circuit design [12], VLSI circuits [13], and information security [14].

The FVS problem is known to be NP-hard for general graphs [15] and bipartite graphs [16]. In general, it is known that more efficient algorithms can be developed by restricting classes of graphs. For instance, interesting polynomial-time solutions for the FVS problem have been found for special classes of graphs, such as interval graphs [17] [18], permutation graphs [19], butterfly networks [20], hypercubes [21], star graphs [22], diamond graphs [23], and rotator graphs [24]. Saha and Pal presented an algorithm that took $O(n+m)$ time for the FVS problem in interval graphs using maximal clique decomposition [18]. The algorithm obtains an MFVS in an interval graph by breaking all cycles for each maximal clique. Circular-arc graphs are a natural generalization of interval graphs. However, the algorithm presented by Saha and Pal [18] cannot be directly applied to circular-arc graphs because the number of maximal cliques in interval graphs is at most the number of vertices, whereas circular-arc graphs may have an exponential number of maximal cliques [25]. In this paper, we propose an algorithm that takes $O(n+m)$ time for the FVS problem in a normal Helly circular-arc graph.

The remainder of this paper is organized as follows. We state the definitions and notations used throughout this paper in Section 2. Next, we present our algorithm for the FVS problem and analyze its complexity in Section 3. Finally, we summarize our findings in Section 4 and conclude the paper by briefly discussing the scope for future work.

2. Definitions and Notations

In this section, we provide the definitions and relevant notations used throughout the paper. These establish the basis of the algorithm presented in Section 3. We provide the definitions of a circular-arc model and its corresponding graph. Consider a unit circle C and a family $\mathcal{F}$ of n arcs $A_1, A_2, \cdots, A_n$ along the circumference of C. Each arc A_i has two endpoints a_i and b_i where a_i (resp., b_i) is the last point encountered when traversing A_i counterclockwise (resp., clockwise). Arc numbers i, j are assigned to each arc in increasing order of their b_i s, *i.e.*, $A_i < A_j$ if $b_i < b_j$. The geometric representation described above is called a circular-arc model. A graph $G=(V,E)$ is called a circular-arc graph if there exists a family of arcs $\mathcal{F}=\{A_1, A_2, \cdots, A_n\}$ such that there is a one-to-one correspondence between vertex $i \in V$ and $A_i \in \mathcal{F}$ such that an edge $(i,j) \in E$ if and only if A_i intersects with A_j in the circular-arc model.

Normal and Helly circular-arc models (NHCM) are precisely those without three or less arcs covering the entire circle [26]. A graph that admits such a model is called a *normal Helly circular-arc graph* (NHCG). Examples of an NHCM and its corresponding graph are shown in **Figure 1**. For an NHCM consisting of n arcs, an arc A_i with $b_n \in A_i$ and $i<n$ is called a *back-arc*. The set of all back-arcs is called the *back-arc set* and is denoted by BA. For CM, shown in **Figure 1(a)**, we have a back-arc set $BA=\{A_1, A_2\}$ by $b_n = b_{12} = 22 \in A_1, A_2$.

A *maximal clique* is a clique to which no further vertices of a graph can be added such that it remains a clique. For the graph G_1 of **Figure 1**, the maximal cliques, the vertices of which are put in ascending order, are $MC_1=\{1,2,3,4\}$, $MC_2=\{1,2,12\}$, $MC_3=\{1,11,12\}$, $MC_4=\{3,4,5\}$, $MC_5=\{4,5,6\}$, $MC_6=\{6,9\}$, $MC_7=\{7,8,9\}$, $MC_8=\{8,9,10\}$, and $MC_9=\{10,11\}$. The cardinality N_j of MC_j for this graph are $N_1=4$, $N_2=N_3=N_4=N_5=N_7=N_8=3$, and $N_6=N_9=2$.

Let r be the number of maximal cliques of NHCG G. Throughout this paper, we use the term *triangle* to denote a cycle whose length is three. We define functions $\sigma(i)$ and $\rho(i)$ as follows:

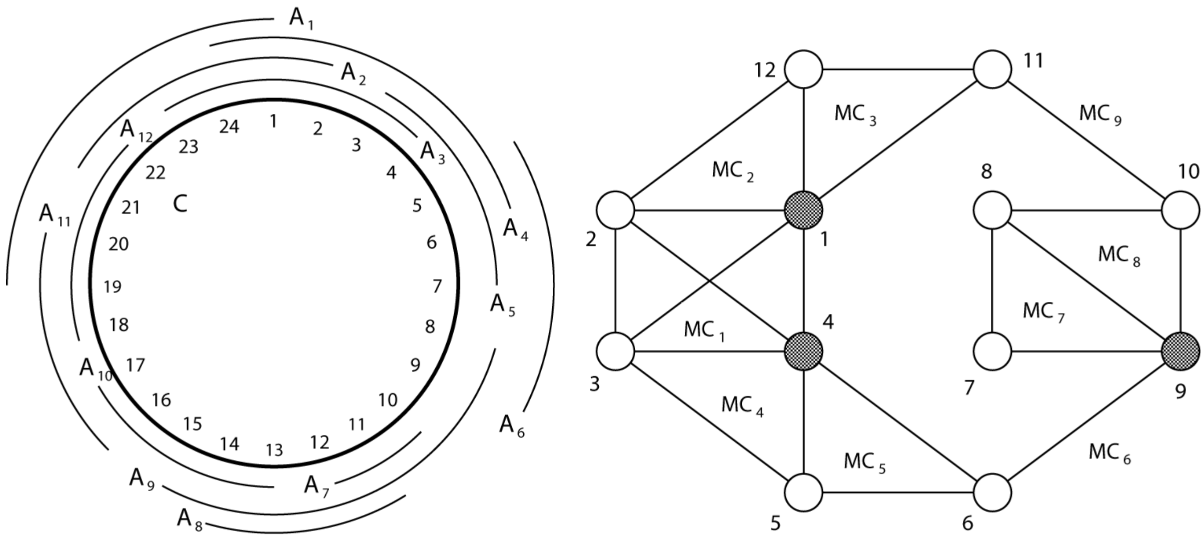

Figure 1. Normal Helly circular-arc model M_1 and its corresponding graph G_1. (a) A normal Helly circular-arc model M_1; (b) A normal Helly circular-arc graph G_1.

$$\sigma(i) = \sum_{MC_j \ni i} \frac{(N_j - 1)(N_j - 2)}{2}, \tag{1}$$

$$\rho(i) = \left| \left\{ MC_j \mid i \in MC_j, N_j = 2, 1 \le j \le r \right\} \right|. \tag{2}$$

Thus, $\sigma(i)$ is the total number of triangles including vertex *i* in *G*. For the sake of convenience, in the example shown in **Figure 1**, we denote the σ and ρ value sequences of *G* by $\sigma = [5,4,4,5,2,1,1,2,2,1,1,2]$ and $\rho = [0,0,0,0,0,1,0,0,1,1,1,0]$, respectively. For a simple graph $G = (V, E)$, $F (\subset V)$ is a *feedback triangle-free vertex set* (FTS) if $G[V - F]$ has no triangle. In the example shown in **Figure 1**, $\{1,4,8\}$ or $\{1,4,9\}$ is a minimum cardinality FTS (MFTS) of *G*.

A chordal graph is a simple graph in which every cycle of length four or greater has a cycle chord. Interval graphs are a subclass of chordal graphs [18]. Hence, an MFTS is obviously an MFVS for interval graphs. On the other hand, NHCGs are a superclass of interval graphs and not a subclass of chordal graphs. They can have some chordless cycles of length greater than three. For example, the graph G_1 shown in **Figure 1** has chordless cycles $\langle 2,3,5,6,9,10,11,12,2 \rangle$ of length eight. If *F* is an MFTS and not an MFVS of an NHCG G_1, $G_1[V - F]$ has a chordless cycle of length greater than three. A chordless cycle in $G_1[V - F]$ is called a *periphery*. For example, in **Figure 1**, $F = \{1,4,8\}$ is an MFTS of G_1, and $G_1[V - F]$ consists of a periphery $\langle 2,3,5,6,9,10,11,12,2 \rangle$. Therefore, $F = \{1,4,8\}$ is not an MFVS, although *F* is an MFTS of G_1.

3. Algorithm and Its Correctness

In this section, we present an algorithm for solving the FVS problem for an NHCG. We will concisely describe the outline of our algorithm. First, we decompose a given NHCG into maximal cliques. An FTS is obtained by removing $N_j - 2$ vertices from each maximal clique MC_j. An MFTS is constructed by minimizing the number of removed vertices. At this point, if the constructed MFTS includes no periphery, it is an MFVS. Otherwise, we can obtain an MFVS by including a vertex for breaking the periphery in the MFTS.

Let $G = (V, E)$ be an NHCG corresponding to a model *M*. Algorithm 1 receives as an input the endpoints a_i, b_i of each A_i and back-arc set *BA*, and outputs an MFVS *F* of *G*.

We use the graph G_1 shown in **Figure 1** as an example to illustrate Algorithm 1 step by step (the updated part is underlined).

Algorithm 1: Algorithm for obtaining MFVS

Input: Endpoints a_i, b_i of A_i and a back-arc set BA.
Output: The minimum feedback vertex set F.

(Step 1)
Compute all maximal cliques MC_j, $1 \leqslant j \leqslant r$;
Compute all $\sigma(i)$ and $\rho(i)$ for $1 \leqslant i \leqslant n$;
(Step 2) /* Compute MFTS */
$U_1 := \{MC_j \mid N_j \geqslant 3, 1 \leqslant j \leqslant r\}$;
$F := \emptyset$;
while $U_1 \neq \emptyset$ **do**
  $k := \arg\max \sigma(i)$;
  $U_2 := \{MC_j \mid k \in MC_j, MC_j \in U_1\}$;
  for *all* $MC_j \in U_2$ **do**
    if $|MC_j - F| \geqslant 3$ **then**
      Select $|MC_j - F| - 2$ vertices in descending lexicographical order with respect to $(\sigma(i), \rho(i))$;
      Add the selected vertices to F ;
    end
    for $i \in MC_j$ **do**
      $\sigma(i) := \sigma(i) - (N_j - 1)(N_j - 2)/2$;
    end
  end
  $U_1 := U_1 - U_2$;
end
(Step 3) /* Break Periphery */
if $G[V - F]$ *has a periphery* **then**
  if $|BA - F| = 1$ $(BA - F = \{v\})$ **then**
    Add vertex v to F ;
  end
  if $|BA - F| = 2$ $(BA - F = \{v_1, v_2\})$ **then**
    Select a vertex v such that $a_k = \min\{a_{v_1}, a_{v_2}\}$;
    Add vertex v to F ;
  end
end

Begin

(Step 1)

$MC_1 = \{1,2,3,4\}$, $MC_2 = \{1,2,12\}$, $MC_3 = \{1,11,12\}$, $MC_4 = \{3,4,5\}$, $MC_5 = \{4,5,6\}$, $MC_6 = \{6,9\}$, $MC_7 = \{7,8,9\}$, $MC_8 = \{8,9,10\}$, and $MC_9 = \{10,11\}$.

$\sigma = [5,4,4,5,2,1,1,2,2,1,1,2]$, $\rho = [0,0,0,0,0,1,0,0,1,1,1,0]$.

(Step 2)

$U_1 = \{MC_1, MC_2, MC_3, MC_4, MC_5, MC_7, MC_8\}$, $F = \varnothing$.

1st iteration

$[k = 1, U_2 = \{MC_1, MC_2, MC_3\}]$

(1-1) $MC_1 = \{1,2,3,4\}$, $F = \{1,4\}$, $\sigma = [\underline{2},\underline{1},\underline{1},\underline{2},2,1,1,2,2,1,1,2]$.

(1-2) $MC_2 = \{1,2,12\}$, $F = \{1,4\}$, $\sigma = [\underline{1},\underline{0},1,2,2,1,1,2,2,1,1,\underline{1}]$.

(1-3) $MC_3 = \{1,11,12\}$, $F = \{1,4\}$, $\sigma = [\underline{0},0,1,2,2,1,1,2,2,1,\underline{0},\underline{0}]$.

$U_1 = \{MC_4, MC_5, MC_7, MC_8\}$.

2nd iteration

$[k = 9, U_2 = \{MC_7, MC_8\}]$

(2-1) $MC_7 = \{7,8,9\}$, $F = \{1,4,9\}$, $\sigma = [0,0,1,2,2,1,\underline{0},\underline{1},\underline{1},1,0,0]$.

(2-2) $MC_8 = \{8,9,10\}$, $F = \{1,4,9\}$, $\sigma = [0,0,1,2,2,1,0,\underline{0},\underline{0},\underline{0},0,0]$.
$U_1 = \{MC_4, MC_5\}$.
3rd iteration
$[k = 4, U_2 = \{MC_4, MC_5\}]$.
(3-1) $MC_4 = \{3,4,5\}$, $F = \{1,4,9\}$, $\sigma = [0,0,\underline{0},\underline{1},\underline{1},1,0,0,0,0,0,0]$.
(3-2) $MC_5 = \{4,5,6\}$, $F = \{1,4,9\}$, $\sigma = [0,0,0,\underline{0},\underline{0},\underline{0},0,0,0,0,0,0]$.
$U_1 = \varnothing$.
(Step 3)
$G[V-F]$ has no periphery. Then, $F := \{1,4,9\}$.
End

In Step 1, all maximal cliques can be generated in $O(n+m)$ time for an NHCG *G* with *n* vertices and *m* edges [27]. Moreover, we compute $\sigma(i)$ and $\rho(i)$ for $1 \le i \le n$. Step 2 constructs an MFTS by adding $N_j - 2$ vertices to *F* for each maximal clique $MC_j\,(N_j \ge 3)$. A graph obtained by deleting all but two vertices from each $MC_j\,(N_j \ge 3)$, $1 \le j \le r$, has no triangle. In the example G_1 shown in **Figure 1**, we first select vertices "1" and "4" and add them to *F* because $\sigma(1) = \sigma(4) = 5$ are the maximum values of all σ values. In the next step, for MC_7, we select vertex "9" and add it to *F* because $\sigma(8) = \sigma(9) = 2$ are maximum values and $\rho(8) = 0$ and $\rho(9) = 1$. $F = \{1,4,9\}$, which we obtained after executing Step 2 of Algorithm 1, is an MFTS of G_1. In Step 3, we check whether $G_1[V-F]$ has a periphery or not. We obtain an MFVS $F = \{1,4,9\}$ because $G_1[V-F]$ has no periphery.

Lemma 1. *Let G be an NHCG. Following the execution of Step* 2 *of Algorithm* 1, *F is an MFTS of G.*

Proof: Each triangle contained in *G* is a subset of any maximal clique $MC_j\,(N_j \ge 3)$ in *G*. A graph obtained by deleting all but two vertices from each maximal clique $MC_j\,(N_j \ge 3)$ has no triangle. Thus, a set *F* consisting of $N_j - 2$ vertices of each $MC_j\,(N_j \ge 3)$ is an FTS of *G*. It is obvious that the cardinality of *F* can be reduced by including vertices that appear in many triangles in *G*. $\sigma(i)$ is, by definition, the total number of triangles including vertex *i* in *G*. An MFTS can be obtained by selecting $N_j - 2$ vertices in descending order of σ for each MC_j.

In Step 2, initially, we set $U_1 = \{MC_j \mid N_j \ge 3, 1 \le j \le r\}$ and $F = \varnothing$. Next, we compute $k = \arg\max \sigma(i)$ and set $U_2 = \{MC_j \mid k \in MC_j, MC_j \in U_1\}$, *i.e.*, vertex *k* is contained in the largest number of triangles, and U_2 is the set of maximal cliques containing *k*. Therefore, we break all triangles containing *k* in $MC_j \in U_2$ by priority to reduce the cardinality of *F*. Here, we assume that U_2 consists of *m* maximal cliques, *i.e.* $U_2 = \{MC_1, MC_2, \cdots, MC_m\}$.

In (1-1) of Step 2, we select all vertices except two minima with σ values in MC_1 and add them to *F* for $MC_1 \in U_2$. Then, a subgraph $G[MC_1 - F]$ has no triangle and *F* is clearly an MFTS of $G[MC_1]$. Because all triangles in MC_1 are broken by removing vertices in *F*, $\sigma(i)$ values, $i \in MC_1$, are updated to $\sigma(i) - (N_1 - 1)(N_1 - 2)/2$.

In (1-2) of Step 2, if $|MC_2 - F| < 3$, no vertex is added to *F*. This implies that the elimination of vertices in *F* obtained in the previous step breaks all triangles of MC_2. If $|MC_2 - F| \ge 3$, we select $|MC_2 - F| - 2$ vertices in decreasing order of σ in $MC_2 - F$ and add them to *F*. It is obvious that the cardinality of *F* can be reduced by adding vertices that appear in several triangles in *F*. Following this step, $G[MC_1 \cup MC_2 - F]$ has no triangle and *F* is an MFTS of $G[MC_1 \cup MC_2]$. Moreover, $\sigma(i)$ values, $i \in MC_2$, are updated to $\sigma(i) - (N_1 - 1)(N_1 - 2)/2$.

Similarly, in the next step (1-3), if $|MC_3 - F| \ge 3$, we select $|MC_3 - F| - 2$ vertices in decreasing order of σ in $MC_3 - F$ and add them to *F*. $G[MC_1 \cup MC_2 \cup MC_3 - F]$ has no triangle, and *F* is an MFTS of $G[MC_1 \cup MC_2 \cup MC_3]$. Using a similar argument, following the execution of the *m*-th step, $G[MC_1 \cup MC_2 \cup \cdots \cup MC_m - F]$ has no triangle, and *F* is an MFTS of $G[MC_1 \cup MC_2 \cup \cdots \cup MC_m]$.

In the second iteration, we update U_1 to be $U_1 - U_2$ and calculate $k = \arg\max \sigma(i)$. As in the case of the first iteration, we select all vertices except two minima with σ values in MC_j and add them to *F* for each $MC_j \in U_2$. Step 2 of Algorithm 1 repeats the processes described above until U_1 becomes an empty set. The method described above thus constructs an MFTS *F* of the NHCG *G*.

Here, we explain how $\rho(i)$ is used to find an MFTS in Step 2 of Algorithm 1. In the example shown in **Figure 1**, both vertex sets $\{1,4,8\}$ and $\{1,4,9\}$ are MFTSs of G_1. In general, not all MTFSs of an NHCG are its MFVSs. For example, a set $\{1,4,9\}$ is an MFVS of G_1; however, it is not an MFVS because a subgraph $G_1[V - \{1,4,8\}]$ has a periphery $\langle 2,3,5,6,9,10,11,12,2 \rangle$. We describe how Step 2 of Algorithm 1 constructs an MFTS F such that $G[V-F]$ has no periphery, if possible.

In Step 2, we select $|MC_j - F| - 2$ vertices from MC_j to break all triangles in MC_j and add them to F. Consider the case where a maximal clique $MC_j(N_j \geq 3)$ and a periphery have a vertex v in common. Clearly, a periphery is broken by removing a vertex v ("5" in **Figure 2(a)**). Such vertex v containing MC_j and a periphery in common must be included in some $MC_k(N_k = 2)$ ($\{1,5\}$, $\{5,6\}$ in **Figure 2(a)**). This implies that $\rho(v) \geq 1$. Moreover, there is no maximal clique $MC_k(N_k = 2)$ containing vertices u ("2", "3", "4" in **Figure 2(a)**) except v in MC_j. This is because it is clear from the corresponding model that if there exists a vertex x adjacent to u $(\neq v)$ in MC_j, G must have a triangle $\langle uvx \rangle$.

Next, we consider the case of a maximal clique $MC_j(N_j \geq 3)$ and a periphery have two vertices v, w in common. It is obvious that a periphery is broken by removing either v or w ("2", "5" in **Figure 2(b)**). In this case, for v and w, there exist maximal cliques $MC_k(N_k = 2)$ ($\{1,2\}$ and $\{5,6\}$ in **Figure 2(b)**) containing v and w, respectively. Moreover, we have no maximal clique $MC_k(N_k = 2)$ containing vertex u, except v and w ("3", "4" in **Figure 2(b)**) in MC_j. This is because it is clear from the corresponding model that if there exists a vertex x adjacent to $u(\neq v, w)$ in MC_k, G must have a triangle $\langle uvx \rangle$ or $\langle uwx \rangle$. Therefore, in Step 2, we select $|MC_j - F| - 2$ vertices in lexicographical order with respect to $(\sigma(i), \rho(i))$ where $(\sigma(i), \rho(i)) > (\sigma(j), \rho(j))$ if $\sigma(i) > \sigma(j)$ or $\sigma(i) = \sigma(j)$, $\rho(i) > \rho(j)$. By executing the method described above, Step 2 outputs an MFTS F such that an NHCG $G[V-F]$ has neither a triangle nor a periphery, if possible.

Thus far, we have presented an example where an MFTS of an NHCG G is also its MFVS. However, there exist cases where an MFTS of G obtained by executing Step 2 of Algorithm 1 is not an MFVS of G. We describe the procedure to construct an MFTS of NHCG G_2 shown in **Figure 3** by executing Step 3.

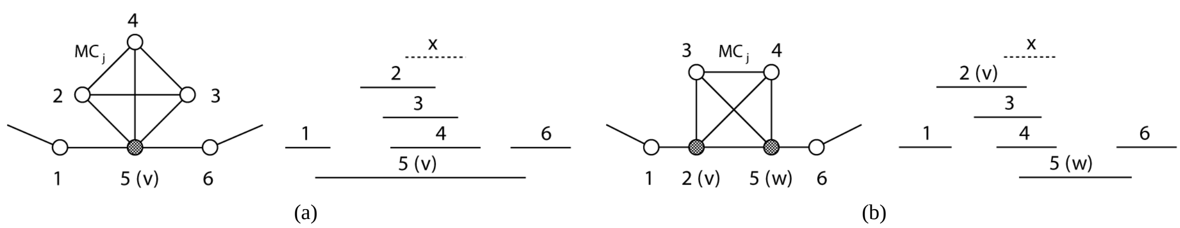

Figure 2. A maximal clique and a periphery sharing vertices. (a) MC shares a vertex with a periphery; (b) MC shares two vertices with a periphery.

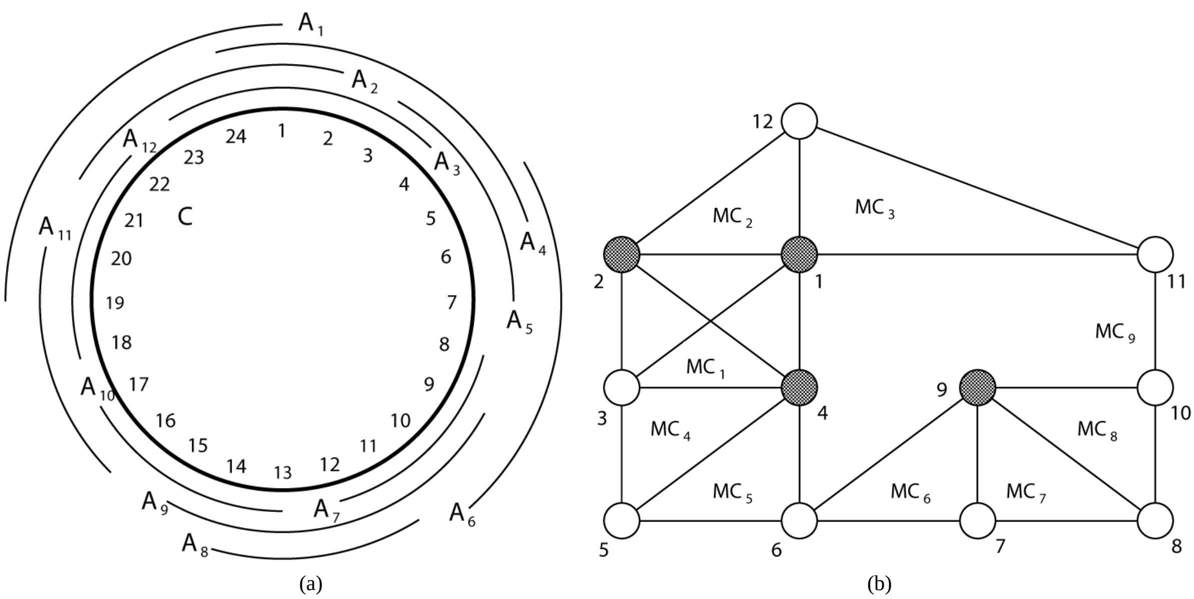

Figure 3. Normal Helly circular-arc model M2 and its corresponding graph G2. (a) A normal Helly circular-arc model M_2; (b) A normal Helly circular-arc graph G_2.

Begin

(Step 1)

$MC_1 = \{1,2,3,4\}$, $MC_2 = \{1,2,12\}$, $MC_3 = \{1,11,12\}$, $MC_4 = \{3,4,5\}$, $MC_5 = \{4,5,6\}$, $MC_6 = \{6,7,9\}$, $MC_7 = \{7,8,9\}$, $MC_8 = \{8,9,10\}$, and $MC_9 = \{10,11\}$.

$\sigma = [5,4,4,5,2,2,2,2,3,1,1,2]$, $\rho = [0,0,0,0,0,0,0,0,0,1,1,0]$.

(Step 2)

$U_1 = \{MC_1, MC_2, MC_3, MC_4, MC_5, MC_7, MC_8\}$, $F = \varnothing$.

1st iteration

$[k = 1, U_2 = \{MC_1, MC_2, MC_3\}]$

(1-1) $MC_1 = \{1,2,3,4\}$, $F = \{1,4\}$, $\sigma = [\underline{2},\underline{1},\underline{1},\underline{2},2,2,2,2,3,1,1,2]$.

(1-2) $MC_2 = \{1,2,12\}$, $F = \{1,4\}$, $\sigma = [\underline{1},\underline{0},1,2,2,2,2,2,3,1,1,\underline{1}]$.

(1-3) $MC_3 = \{1,11,12\}$, $F = \{1,4\}$, $\sigma = [\underline{0},0,1,2,2,2,2,2,3,1,\underline{0},\underline{0}]$.

$U_1 = \{MC_4, MC_5, MC_6, MC_7, MC_8\}$.

2nd iteration

$[k = 9, U_2 = \{MC_7, MC_8\}]$

(2-1) $MC_6 = \{6,7,9\}$, $F = \{1,4,9\}$, $\sigma = [0,0,1,2,2,\underline{1},\underline{1},2,\underline{2},1,0,0]$.

(2-2) $MC_7 = \{7,8,9\}$, $F = \{1,4,9\}$, $\sigma = [0,0,1,2,2,1,\underline{0},\underline{1},\underline{1},1,0,0]$.

(2-3) $MC_8 = \{8,9,10\}$, $F = \{1,4,9\}$, $\sigma = [0,0,1,2,2,1,0,\underline{0},\underline{0},\underline{0},0,0]$.

$U_1 = \{MC_4, MC_5\}$.

3rd iteration

$[k = 4, U_2 = \{MC_4, MC_5\}]$

(3-1) $MC_4 = \{3,4,5\}$, $F = \{1,4,9\}$, $\sigma = [0,0,\underline{0},\underline{1},\underline{1},1,0,0,0,0,0,0]$.

(3-2) $MC_5 = \{4,5,6\}$, $F = \{1,4,9\}$, $\sigma = [0,0,0,\underline{0},\underline{0},\underline{0},0,0,0,0,0,0]$.

$U_1 = \varnothing$.

(Step 3)

$G[V-F]$ has a periphery $\langle 2,3,5,6,7,8,10,11,12,2 \rangle$.

We have $F := \{1,4,9\} \cup \{2\}$ by $|BA - F| = 1$.

End

Following the execution of Step 2 of Algorithm 1, we obtain an MFTS $F = \{1,4,9\}$ of G_2. Removing all vertices in $F = \{1,4,9\}$ breaks all triangles in G_2. However, *F* is not an MFVS of G_2 because $G_2[V-F]$ has a periphery $\langle 2,3,5,6,7,8,10,11,12,2 \rangle$. The periphery remains unbroken because it does not contain any of $F = \{1,4,9\}$. In fact, there exists no MFVS of cardinality three in G_2. Hence, we can obtain an MFVS by adding a vertex for breaking the periphery to *F* if $G[V-F]$ consists of a periphery. In Step 3, we include vertex '2' in *F* by $|BA - F| = 1$ and $BA - F = \{2\}$. We can obtain an MFVS $F = \{1,2,4,9\}$ of G_2.

The following lemmas guarantee the validity of Algorithm 1.

Lemma 2. *Let G be a normal Helly circular-arc graph. If F is an MFTS and not an MFVS of G, a periphery in* $G[V-F]$ *must contain one or two back-arcs.*

Proof: As mentioned in Section 2, interval graphs are a subclass of NHCGs and have no periphery. An NHCM from which all back-arcs are removed is topologically equivalent to an interval model. Therefore, $G[V-F]$ must contain at least one back-arc because it has a periphery.

$G[V-F]$ does not contain three or more back-arcs. If $G[V-F]$ has three back-arcs, these three back-arcs cover a point b_n on the circumference of a circle *C* by the definition of a back-arc. This implies that $G[V-F]$ contains a triangle. Thus, it contradicts the proposition that *F* is an MFTS of *G*.

Thus, if *F* is an MFTS and not an MFVS of G, $G[V-F]$ has a periphery that must contain one or two back-arcs.

Lemma 3. *Let G be a normal Helly circular-arc graph. Following the execution of Step* 3 *of Algorithm* 1, *F is an MFVS of G*.

Proof: By Lemma 2, if $G[V-F]$ consists of a periphery after executing Step 2, such a periphery must contain one or two back-arcs.

In the case where $G[V-F]$ has a periphery and contains one back-arc A_1, we can break the periphery by removing A_1 (**Figure 4(a)**). Thus, we can obtain an MFVS by adding A_1 to *F*.

We consider the cases where $G[V-F]$ has a periphery and contains two back-arcs A_1, A_2 ($b_1 < b_2$). Because both back-arcs cover point b_n by definition of back-arc, these two back-arcs must intersect. There are

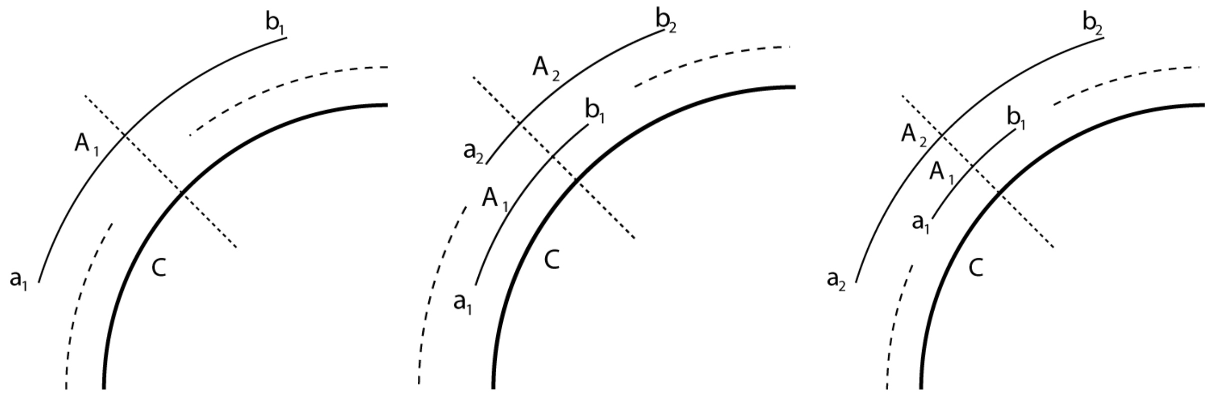

Figure 4. Illustration of Lemma 3. (a) $|BA-F|=1$; (b) $a_1<a_2$, (c) $a_2<a_1$.

two possible cases where $G[V-F]$ contains two back-arcs. The first satisfies $a_1<a_2$ (**Figure 4(b)**), and the second satisfies $a_1>a_2$ (**Figure 4(c)**). For the former, the periphery is broken by removing A_1. For the latter, the periphery is broken by removing A_2. Therefore, we can break the periphery by removing a back-arc A_k such that $a_k=\min\{a_1,a_2\}$.

Therefore, we can construct an MFVS after executing Step 3 of Algorithm 1.

In the following, we analyze the complexity of Algorithm 1. In Step 1, all maximal cliques of *G* are computed in $O(n+m)$ time [27]. Moreover, $\sigma(i)$ and $\rho(i)$ are computed for all $i\in V$. Its complexity depends on the number of maximal cliques of *G*, which is at most the number of vertices of *G*. In Step 2, an MFTS *F* of *G* is constructed. This step requires as many iterations as the number of maximal cliques. Thus, this step is executed in $O(n)$ time. In Step 3, we check whether $G[V-F]$ consists of a periphery. If $G[V-F]$ consists of a periphery, one vertex of $BA-F$ is added to *F*. This step is executed in $O(n)$ time. Thus, we have the following theorem.

Theorem 1. *Algorithm* 1 *finds an MFVS of a normal Helly circular-arc graph G in* $O(n+m)$ *time.*

4. Concluding Remarks

In this paper, we proposed an algorithm that takes $O(n+m)$ time to find an MFVS on an NHCG with *n* vertices and *m* edges. Our algorithm employs other algorithm to find maximal cliques [27] according to a method that can be understood intuitively. The complexity of our algorithm depends on the number of maximal cliques in an NHCG. Reducing the complexity of the algorithm and extending the results to other graphs are issues to be explored in future research.

Acknowledgements

We thank the Editor and the referee for their comments. This work was partially supported by JSPS KAKENHI Grant Number 25330019.

References

[1] Golumbic, M.C. (2004) Algorithmic Graph Theory and Perfect Graphs. 2nd Edition, Vol. 57, Elsevier.

[2] Stahl, F.W. (1967) Circular Genetic Maps. *Journal of Cellular Physiology*, **70**, 1-12. http://dx.doi.org/10.1002/jcp.1040700403

[3] Hubert, L. (1974) Some Applications of Graph Theory and Related Non-Metric Techniques to Problems of Approximate Seriation: The Case of Symmetric Proximity Measures. *British Journal of Mathematical and Statistical Psychology*, **27**, 133-153. http://dx.doi.org/10.1111/j.2044-8317.1974.tb00534.x

[4] Tucker, A. (1975) Coloring a Family of Circular-Arcs. *SIAM Journal on Applied Mathematics*, **29**, 493-502. http://dx.doi.org/10.1137/0129040

[5] Stefanakos, S. and Erlebach, T. (2004) Routing in All-Optical Ring Networks Revisited. *Proc.* 9*th IEEE Symposium on*

Computers and Communication, **1**, 288-293. http://dx.doi.org/10.1109/iscc.2004.1358419

[6] Lin, M.C., Soulignac, F.J. and Szwarcfiter, J.L. (2010) The Clique Operator on Circular-Arc Graphs. *Discrete Applied Mathematics*, **158**, 1259-1267. http://dx.doi.org/10.1016/j.dam.2009.01.019

[7] Bhowmick, D. and Chandran, L.S. (2011) Boxicity of Circular Arc Graphs. *Graphs and Combinatorics*, **27**, 769-783. http://dx.doi.org/10.1007/s00373-010-1002-1

[8] Kaplan, H. and Nussbaum, Y. (2011) A Simpler Linear-Time Recognition of Circular-Arc Graphs. *Algorithmica*, **61**, 694-737. http://dx.doi.org/10.1007/s00453-010-9432-y

[9] Cerioli, M.R. and Korenchendler, A.L. (2009) Clique-Coloring Circular-Arc Graphs. *Electronic Notes in Discrete Mathematics*, **35**, 287-292. http://dx.doi.org/10.1016/j.endm.2009.11.047

[10] Jungnickel, D. (2000) Graphs, Networks and Algorithms. Springer.

[11] Silberschatz, A., Galvin, P.B. and Gagne, G. (2003) Operating Systems Concepts. John Wiley and Sons, New York.

[12] Johnson, D.B. (1975) Finding All the Elementary Circuits of a Directed Graph. *SIAM Journal on Computing*, **4**, 77-84. http://dx.doi.org/10.1137/0204007

[13] Hudli, A.V. and Hudli, R.V. (1994) Finding Small Feedback Vertex Sets for VLSI Circuits. *Microprocessors and Microsystems*, **18**, 393-400. http://dx.doi.org/10.1016/0141-9331(94)90067-1

[14] Gusfield, D. (1998) A Graph Theoretic Approach to Statistical Data Security. *SIAM Journal on Computing*, **17**, 552-571. http://dx.doi.org/10.1137/0217034

[15] Garey, M.R. and Johnson, D.S. (1979) Computers and Intractability: A Guide to Theory of NP-Completeness, W. H. Freeman, San Francisco.

[16] Yannakakis, M. (1981) Node-Deletion Problem on Bipartite Graphs. *SIAM Journal on Computing*, **10**, 310-327. http://dx.doi.org/10.1137/0210022

[17] Lu, C.L. and Tang, C.Y. (1997) A Linear-Time Algorithm for the Weighted Feedback Vertex Problem on Interval Graphs. *Information Processing Letters*, **61**, 107-111. http://dx.doi.org/10.1016/S0020-0190(96)00193-7

[18] Saha, A. and Pal, M. (2005) An Algorithm to Find a Minimum Feedback Vertex Set of an Interval Graph. *Advanced Modeling and Optimization*, **7**, 99-116.

[19] Liang, Y.D. (1994) On the Feedback Vertex Set Problem in Permutation Graphs. *Information Processing Letters*, **52**, 123-129. http://dx.doi.org/10.1016/0020-0190(94)00133-2

[20] Luccio, F.L. (1998) Almost Exact Minimum Feedback Vertex Set in Meshes and Butterflies. *Information Processing Letters*, **66**, 59-64. http://dx.doi.org/10.1016/S0020-0190(98)00039-8

[21] Focardi, R., Luccio, F.L. and Peleg, D. (2000) Feedback Vertex Set in Hypercubes. *Information Processing Letters*, **76**, 1-5. http://dx.doi.org/10.1016/S0020-0190(00)00127-7

[22] Wang, F.H., Wang, Y.L. and Chang, J.M. (2004) Feedback Vertex Sets in Star Graphs. *Information Processing Letters*, **89**, 203-208. http://dx.doi.org/10.1016/j.ipl.2003.11.001

[23] Carrabs, F., Cerulli, R., Gentili, M. and Parlato, G. (2005) A Linear Time Algorithm for the Minimum Weighted Feedback Vertex Set on Diamonds. *Information Processing Letters*, **94**, 29-35. http://dx.doi.org/10.1016/j.ipl.2004.12.008

[24] Kuo, C.J., Hsu, C.C., Lin, H.R. and Lin, K.K. (2009) An Efficient Algorithm for Minimum Feedback Vertex Sets in Rotator Graphs. *Information Processing Letters*, **109**, 450-453. http://dx.doi.org/10.1016/j.ipl.2009.01.004

[25] Tucker, A. (1980) An Efficient Test for Circular-Arc Graphs. *SIAM Journal on Computing*, **9**, 1-24. http://dx.doi.org/10.1137/0209001

[26] McKee, T.A. (2003) Restricted Circular-Arc Graphs and Clique Cycles. *Discrete mathematics*, **263**, 221-231. http://dx.doi.org/10.1016/S0012-365X(02)00578-2

[27] Pal, M. and Bhattacharjee, G.P. (1995) An Optimal Parallel Algorithm for Computing All Maximal Cliques of an Interval Graph and Its Applications. *Journal of The Institution of Engineers* (*India*), **76**, 22-33.