AMO - Advanced Modeling and Optimization, Volume 8, Number 1, 2006

An Optimal Algorithm to Solve 2-Neighbourhood Covering Problem on Circular-arc Graphs

Swagata Mandal and Madhumangal Pal [1]

Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore – 721 102, India.
e-mail: madhumangal@lycos.com

Abstract. Let $G = (V, E)$ be a simple graph and k be a fixed positive integer. A vertex w is said to be a k-neighbourhood cover of an edge (u, v) if $d(u, w) \leq k$ and $d(v, w) \leq k$. A set $C \subseteq V$ is called a k-neighbourhood-covering set if every edge in E is k-neighbourhood covered by some vertices of C. The minimum k-neighbourhood covering problem is to find a set $C \subseteq V$ such that cardinality of C is minimum among all k-neighbourhood covering sets. This problem is NP-complete for general graphs also it remains NP-complete for chordal graphs. An $O(n)$ time algorithm is designed to solve minimum 2-neighbourhood-covering problem on a circular-arc graph. A data structure called interval tree is used to solve this problem.

Keywords: Design and analysis of algorithms, 2-neighbourhood-covering, circular-arc graph, interval graph, interval tree.

AMS Subject Classifications: 68Q22, 68Q25, 68R10.

1 Introduction

A graph $G = (V, E)$ is called an intersection graph for a finite family $\mathcal{F}$ of a non empty set if there is a one-to-one correspondence between $\mathcal{F}$ and V such that two sets in $\mathcal{F}$ have non empty intersection if and only if their corresponding vertices in V are adjacent to each other. $\mathcal{F}$ is called an intersection model of G and G is called the intersection graph of $\mathcal{F}$. If $\mathcal{F}$ is a family of arcs around a circle, then G is called a circular-arc graph. If $\mathcal{F}$ is a family of line segments on real line, then G is called an interval graph. V is the set of all vertices and E is the set of all edges of the graph G.

Circular-arc graph is a general form of interval graph [4, 9] and it is one of the most useful discrete mathematical structure for modelling problems arising in the real world. It has many applications in genetics, traffic control, cyclic scheduling and computer compiler design.

Turker [14] has proposed $O(n^3)$ time algorithm for recognizing a circular-arc graph and constructing in the affirmative case, a circular arc model. Hsu [5] has designed an $O(nm)$ time algorithm for this problem. Eschen and Spinrad [3] have presented an $O(n^2)$ time algorithm for recognizing a circular-arc graph.

[1] This work has been done as a part of the project sponsored to the second author by Department of Science and Technology, India, under grant No. SR/FTP/ETA-008/2002.

In a graph $G = (V, E)$, the *length* of a path is the number of edges in the path. The *distance* $d(x, y)$ from the vertex x to the vertex y is the minimum length of a path from x to y, and if there is no path from x to y then $d(x, y) = \infty$.

A vertex x k-dominates another vertex y if $d(x, y) \leq k$. A vertex z k-neighbourhood-covers (k-NC) an edge (x, y) if $d(x, z) \leq k$ and $d(y, z) \leq k$ i.e., the vertex z k-dominates both the vertices x and y. Conversely, if $d(x, z) \leq k$ and $d(y, z) \leq k$ then the edge (x, y) is said to be k-neighbourhood covered by the vertex z. A set of vertices $C \subseteq V$ is a k-NC set if every edge in E is k-NC by some vertex in C. The k-NC number $\rho(G, k)$ is the minimum cardinality of all k-NC set.

The k-neighbourhood-covering (k-NC) problem is a variant of the domination problem. Domination is a natural model for location problems in operations research, networking etc.

The graphs, considered in this paper are simple i.e., finite, undirected and having no self-loop or parallel edges. For $k = 1$, Lehel et al. [7] have presented a linear time algorithm for computing $\rho(G, 1)$ for an interval graph G. Chang et al. [1] and Hwang et al. [6], have presented linear time algorithms for computing $\rho(G, 1)$ for a strongly chordal graph G provided that strong elimination ordering is known. Hwang et al. [6] have also proved that (k-NC) problem is NP-complete for chordal graphs. In [8], Mondal et al. have designed an optimal algorithm for finding 2-NC set on interval graphs, and their algorithm take $O(n)$ time.

In this paper, an $O(n)$ time algorithm is designed to solve minimum 2-neighbourhood-covering problem on circular-arc graphs. A data structure called interval tree (IT) [10, 11] is used to solve this problem.

2 Definition and Preliminaries

Let $A = \{A_1, A_2, \ldots, A_n\}$ be the circular arc family of circular-arc graph $G = (V, E)$. The family of circular arcs are located around a circle $\mathcal{C}$. While going in a clockwise direction, the point at which we first encounter an arc will be called the *starting point* of the arc. Similarly, the point at which we leave an arc will be called the *finishing point* of that arc. Every arc can be represented by their two endpoints e.g., A_i can be represented as $[s_i, f_i]$, where s_i is the starting point and f_i is the finishing point of the arc A_i on the circle $\mathcal{C}$. Each endpoint of an arc is assigned to a positive integer called a *coordinate*. A *ray* is a straight line from the centre of $\mathcal{C}$ passing through any coordinate. A *path* of a graph G is an alternating sequence of distinct vertices and edges, beginning and ending with vertices. The length of a path is the number of edges in the path. A path from vertex i to j is a *shortest path* if there is no other path from i to j with lower length. The shortest distance (i.e., the length of the shortest path) between the vertices i and j is denoted by $d(i, j)$.

We consider a ray through starting point of any arc. Then, consider the arcs which are intersected by the ray. Find out the arc which has right most finishing point among the arcs which are intersected by the ray. We label this arc by n , then start anticlockwise traversal from the finishing point of the arc which is labelled n. We label $(n-1)$ to the arc with next successive finishing point. In this process, we label all the remaining arcs.

Without loss of generality, we assume the following :

1. An arc contains both its end points and that no two arcs share a common end point.
2. The graph G is connected and the list of sorted endpoints are given.
3. No single arc in A cover the entire circle $\mathcal{C}$.

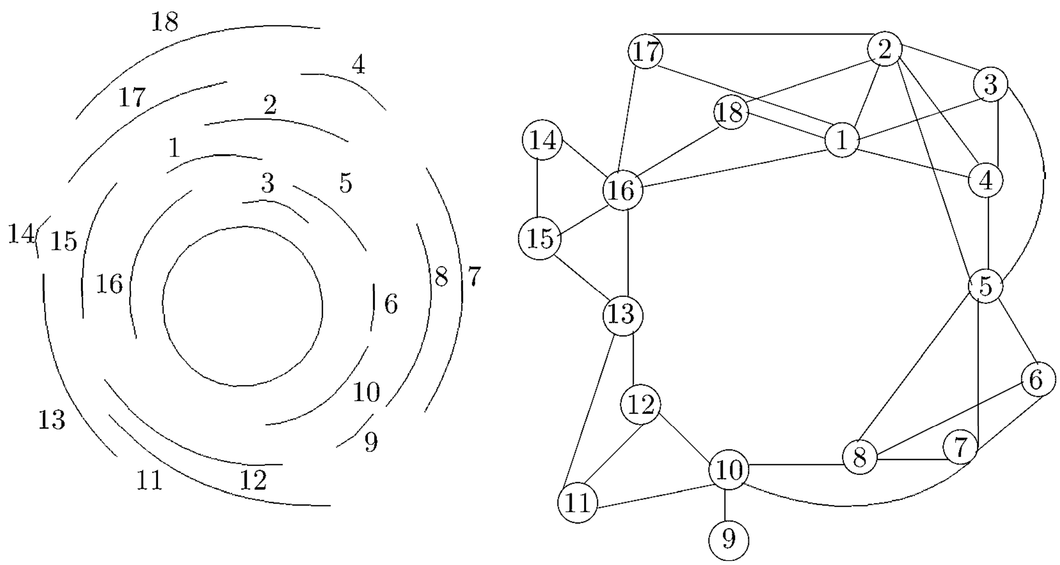

Figure 1: Example of a circular-arc graph and its circular arc representation

4. Arcs and vertices of a circular-arc graph are same thing.
5. The endpoints of the arcs in A are sorted according to the order in which they are visited during the anticlockwise traversal along circle by starting at an arbitrary arc called A_n.
6. The arcs are sorted in decreasing values of f_i's i.e., $f_i < f_j$ for $i < j$.
7. $\bigcup_{i=1}^{n} A_i = \mathcal{C}$ (otherwise, the problem becomes one on interval graph).

The family of arcs A is said to be canonical if
(i) s_i's and f_i's for all $i = 1, 2, \ldots, n$ are distinct integers between 1 to $2n$ and
(ii) point $2n$ is the finishing end point of the arc A_n.

If A is not canonical, using sorting one can construct a canonical family of arcs using $O(n \log n)$ time.

3 Representation of a Circular-arc Graph as Interval Graph

The 2-neighbourhood covering problem on circular-arc graph is solved by converting it to an appropriate interval graph. The main reason for this conversion is that the interval graph can be easily take up with its good data structure interval tree. Pal and Bhattachajee [10] have developed the data structure interval tree and Pal et al. have several problems on interval graphs and also on circular-arc graphs [10, 11, 12, 13]. Thus to solve the problem we first transfer the family of arcs to an equivalent family of intervals on a real line.

Let A be the set of arcs of the circular-arc graph and I be the set of intervals on the real line. First, we consider a ray through the finishing point of the arc A_n i.e., f_n. We consider the arc A_n as an interval I_n, where finishing endpoint of A_n is right endpoint of I_n and starting endpoint of A_n is left endpoint of I_n. Similarly, we transfer all arcs $A_i, i = 1, 2, \ldots, n-1$ of the circular-arc graph G to the interval I_i of the interval graph. Also we add one more interval corresponding to arc A_n and we label this interval as 0. The left endpoint of the interval I_0 is

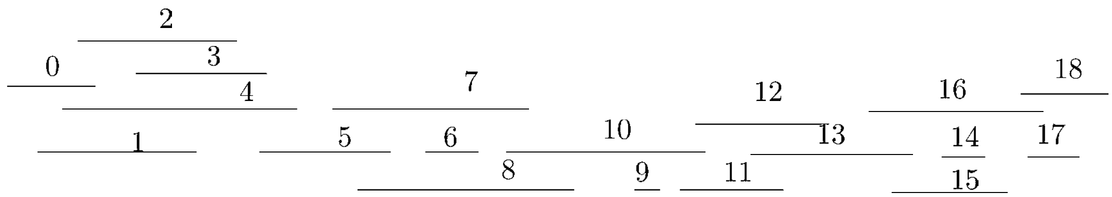

Figure 2: The family of intervals corresponding to the family of arcs of Figure 1

less than the left endpoint of the interval I_1 and the right endpoint of I_0 is greater than the left endpoint of the interval I_1.

We define the interval graph corresponding to the circular-arc graph G as $G' = (V', E')$. In G', there is one more vertex corresponding to the interval I_0. So, we define the vertex set of interval graph as V' which is equal to $V \bigcup \{0\}$.

The interval representation of the graph of Figure 1 is given in Figure 2.

Interval tree is used as a data structure to develop the algorithm to solve the 2-NC problem. Thus, a brief introduction is given below, details available in [10].

4 Properties of Interval tree

In this section, we make use of particular characterization of interval graph that was mentioned in [2]. Here the interval graph is $G' = (V', E')$ and there is a linear order '$<$' on the set of vertices V'.

Lemma 1 *If the vertices $u, v, w \in V'$ are such that $u < v < w$ in the '$<$' ordering and u is adjacent to w, then v is also adjacent to w. But v is not necessarily adjacent to u.*

Such an ordering of vertices is said to be *umbrella free*. In particular, if the graph is given as a collection of intervals, the ordering of interval right bound positions satisfies this property.

For each vertex $v \in V'$ let $H(v)$ be the highest numbered adjacent vertex of v. If there is no vertex adjacent to v and greater than v then $H(v)$ is assumed to be v. In other words

$$H(v) = max\{u : (u, v) \in E',\ u \geq v\}.$$

The array $H(v), v \in V'$ satisfies the following important result.

Lemma 2 *[10] If $u, v \in V'$ and $u < v$ then $H(u) \leq H(v)$.*

For an interval graph $G' = (V', E')$, the interval tree (IT) with root n be defined as $T(G') = (V', E'')$ where $E'' = \{(u, v) : u \in V'$ and $v = H(u), u \neq n\}$.

In [10], it is proved that for a connected interval graph there exists a unique interval tree.

The interval tree $T(G')$ of the interval graph of Figure 2 is shown in Figure 3.

Since the tree $T(G')$ is built from the vertex set V' and the edge set $E'' \subseteq E'$, $T(G')$ is a spanning tree of G'. Let N_k be the set of vertices which are at a distance k from the vertex n in IT. Thus $N_k = \{u :\ d(u, n) = k\}$ and N_0 is the singleton set $\{n\}$.

For each vertex u of IT, we define *level* of u to be the distance of u from the vertex n in the tree IT i.e., $level(u) = d(u, n)$. If $u \in N_k$ then $d(u, n) = k$ and the vertex u is at level k of IT. Thus the vertices at level k of IT are the vertices of N_k.

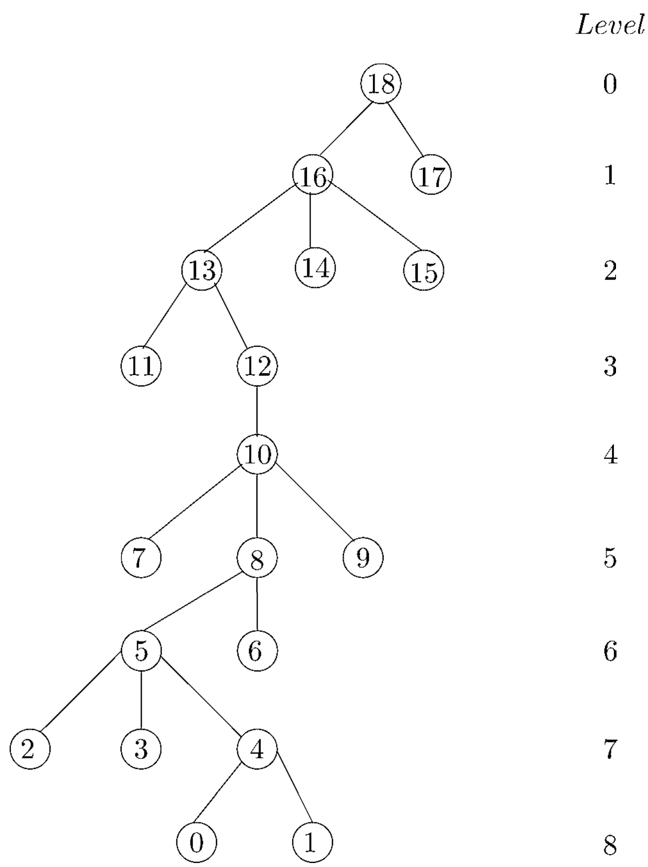

Figure 3: Interval tree of the interval graph of Figure 2

The property that the vertices at any level of IT are the consecutive integers, is proved in [10] as the following lemma.

Lemma 3 *[10] The vertices of N_k are consecutive integers and if v is equal to $min\{u : u \in N_k\}$, then $max\{u : u \in N_{k+1}\}$ is equal to $v - 1$.*

The following result is also proved in [10].

Lemma 4 *If $level(u) < level(v)$ then $u > v$.*

If the level of a vertex v of IT is k then it should be adjacent only to the vertices at levels $k-1$, k and $k+1$ in G'. This observation is proved in [10] as following lemma.

Lemma 5 *If $u, v \in V'$ and $|level(v) - level(u)| > 1$ then $(u, v) \notin E'$.*

The distance $d(u, v)$ between any two vertices u and v of same level is either 1 or 2, which is proved in [10] as follows.

Lemma 6 *[10] For $u, v \in V'$ if $level(v) = level(u)$ then*

$$d(u,v) = \begin{cases} 1, & (u,v) \in E' \\ 2, & \text{otherwise.} \end{cases}$$

Let the notation $u \to v$ be used to indicate, that there is a path from u to v of the length one.

The path in IT from the vertex 0 to the root n is called *main path*. Throughout the paper, we denote the vertex at level l on the main path by u_l^* for all l. From the definition of IT and its level it is obvious that $level(0)$ is equal to the height (h') of the tree IT.

5 2-Neighbourhood-Covering Set

Let C be the minimum 2-neighbourhood-covering (2-NC) set of the given circular-arc graph G. We construct a IT rooted at n and denote it by T_n^*. Then we find four vertices $u_1^*, u_2^*, u_3^*, u_4^*$ at levels 1, 2, 3, 4 on the main path of T_n^*. Then we represent four interval graph representations from the circular-arc graph G, where last vertices of the interval graphs are $u_1^*, u_2^*, u_3^*, u_4^*$ respectively. From four interval graphs we construct four interval trees $T_1^*, T_2^*, T_3^*, T_4^*$. Then we find 2-neighbourhood-covering sets $C_1^*, C_2^*, C_3^*, C_4^*$ from each of the interval trees $T_1^*, T_2^*, T_3^*, T_4^*$. Then we identify the set which has minimum cardinality among the sets $C_1^*, C_2^*, C_3^*, C_4^*$. This minimum cardinality set is the 2-NC set of the circular-arc graph G.

First we represent the graph G_1^*. In G_1^*, the vertex u_1^* is taken as the last vertex n. Let I^* be the set of intervals of the graph G_1^*. First, we consider the arc $A_{u_1^*}$ corresponding to the vertex u_1^* as the interval I_n^*, where finishing point of $A_{u_1^*}$ is the right endpoint of the interval I_n^* also the starting point of $A_{u_1^*}$ is the left endpoint of the interval I_n^*. Then we transfer the next consecutive arc in anticlockwise direction of the arc $A_{u_1^*}$ as the interval $I_{(n-1)}^*$. Similarly, we transfer all other arcs of the circular-arc graph to the intervals of the set I^*. Also, we add one more interval I_0^* corresponding to the arc $A_{u_1^*}$, where the left endpoint of I_0^* is less than the left endpoint of I_1^* and right endpoint of I_0^* is greater than the left endpoint of I_1^*. Similarly, we construct the interval graphs G_2^*, G_3^*, G_4^*.

From the interval graph G_1^* we can get an interval tree T_1^* rooted at u_1^*. From the tree T_1^* we find a 2-NC set C_1^* of the circular-arc graph G. But the name of vertex of the tree is different from the original name of the circular-arc graph G. So we consider a number p_i^* such that $p_i^* = (n - u_i^*)$ for all $i = 1, 2, 3, 4$. If the name of any vertex v of the set C_i^* is greater than p_i^* then we subtract p_i^* from v i.e., we take v as $v - p_i^*$. Also if the name of any vertex v of the set C_i^* is less or equal to p_i^* then we add $(n - p_i^*)$ with v i.e., we take v as $v + (n - p_i^*)$. After that we get the original name of the vertices of the circular-arc graph.

Here we introduce some notations which are used throughout the remaining part of the paper.

parent	if $H(u) = v$ then the $parent(u) = v$ in IT.
gparent	if $parent(parent(u)) = v$ then $gparent(u) = v$.
l	an integer representing the level number at any stage.
u_l^*	represent the vertex on the main path at level l.
X_l	the set of vertices at level l of IT which are greater than u_l^* i.e.,

$$X_l = \{v : \ v > u_l^* \text{ and } v \in N_l\}.$$

Y_l	the set of vertices at level l of IT which are less than u_l^* i.e.,

$$Y_l = \{v : \ v < u_l^* \text{ and } v \in N_l\}.$$

w_l	the least vertex of the set Y_l i.e.,

$$w_l = \min\{v : \ v \in Y_l\}.$$

If $d(u_l^*, x) \le 2$ and $d(u_l^*, y) \le 2$ for any vertex u_l^*, then the edge $(x, y) \in E$ is $2NC$ by u_l^*.

It may be noted that $X_l \bigcap Y_l = \Phi$ and $N_l = X_l \bigcup Y_l \bigcup \{u_l^*\}$. As the vertices of N_l are consecutive integers, the vertices of X_l and Y_l are also consecutive integers.

Lemma 7 *The root u_0^* of the tree T_i^* is a possible first member of C_i^*.*

Proof: If the graph is a circular-arc graph then each vertex u_l^* of the main path is $2NC$ by some edges (x, y) where $x \in N_{l-1} \bigcup N_{l-2}$, $y \in N_{l-1} \bigcup N_{l-2}$ and some edges by (x', y') where $x' \in N_{l+1} \bigcup N_{l+2}$, $y' \in N_{l+1} \bigcup N_{l+2}$.

The vertex u_0^* is $2NC$ by all the edges (x, y) where $x \in N_1 \bigcup N_2$, $y \in N_1 \bigcup N_2$. Also, we know u_l^* is the same vertex of the vertex u_0^*. u_l^* is $2NC$ by all the edges (x, y), where $x \in N_{l-1}$ and Y_{l-1}. Also, u_l^* is $2NC$ by some the edges (x', y'), where $x' \in Y_{l-2} \bigcup N_{l-1}$, $y' \in Y_{l-2} \bigcup N_{l-1}$. So, we can take any one vertex of the main path as the first member of the set C_i^*. So the vertex u_0^* is the possible first member of the set C_i^*. □

If u_l^* be selected as a member of $2NC$ set at any stage then in the next stage either u_{l+3}^* or u_{l+4}^* is to be selected as a member of $2NC$ set. The selection depends on some results which are considered below.

Lemma 8 *If v be any member of $\bigcup_{j=0}^{2} X_{j+l}$ the $d(v, u_l^*) \leq 2$.*

Proof: From the definition of X_l it follows that $u_l^* < v$ for all $v \in X_l$ and for all l. If $v \in X_l$ then $level(v) = level(u_l^*)$ and by Lemma 6, $d(v, u_l^*) \leq 2$.

If v_2 be any vertex of X_{l+1} (see Figure 4) then $u_{l+1}^* < v_2 < u_l^*$. Since $(u_{l+1}^*, u_l^*) \in E$ then by Lemma 1 $(v_2, u_l^*) \in E$. So, $d(v_2, u_l^*) = 1$. Let v_1 be any vertex of X_{l+2}. Then $u_{l+2}^* < v_1 < u_{l+l}^*$. Since $(u_{l+2}^*, u_{l+l}^*) \in E$ then by Lemma 1 $(v_1, u_{l+l}^*) \in E$. Therefore, distance of the path $v_1 \rightarrow u_{l+1}^* \rightarrow u_l^*$ is 2 i.e., $d(v_1, u_l^*) = 2$.

Thus $d(v, u_l^*) \leq 2$ for all $v \in \bigcup_{j=0}^{2} X_{j+l}$. □

Lemma 9 *If t be any member of $\bigcup_{j=0}^{2} Y_{j+l}$ then either $d(t, u_l^*) \leq 2$ or $d(t, u_{l+3}^*) \leq 2$.*

Proof: To proved this lemma consider the IT of Figure 4. Let t_1 and t_2 be any two vertices of Y_{l+2} and Y_{l+1} respectively. Let t_3 be any vertex at level l and $t_3 < u_l^*$. There are two cases arise. Case I: $t_3 = u_l^*$ and Case II: $t_3 \neq u_l^*$.

Case I: In this case $d(t_2, u_l^*) = 1$ and $d(t_1, u_l^*) = 2$. Also, by Lemma 6, $d(t, u_l^*) \leq 2$ for all $t \in Y_l$. Therefore, $d(t, u_l^*) \leq 2$ for all $t \in \bigcup_{j=0}^{2} Y_{j+l}$.

Case II: Without loss of generality we assume that $parent(t_1) = t_2$ and $parent(t_2) = t_3$. Since $parent(t_1) = t_2$ i.e., $H(t_1) = t_2 < u_{l+1}^*$, $(t_1, u_{l+1}^*) \notin E$. Similarly, $H(t_2) = t_3 < u_l^*$ implies $(t_2, u_l^*) \notin E$. Thus the distance of the path $t_2 \rightarrow t_3 \rightarrow u_l^*$ is 2 and the distance of the path $t_1 \rightarrow t_2 \rightarrow u_{l+1}^* \rightarrow u_l^*$ is 3. So, $d(u_l^*, t_2) = 2$ and $d(t_1, u_l^*) = 3$.

Now, $u_{l+3}^* < t_1 < u_{l+2}^* < t_2$, $(u_{l+3}^*, u_{l+2}^*) \in E$ and $(t_1, t_2) \in E$ implies $(t_1, u_{l+2}^*) \in E$ and $(t_2, u_{l+2}^*) \in E$. Thus $d(t_1, u_{l+3}^*) \leq 2$ and $d(t_2, u_{l+3}^*) = 2$. Hence, either $d(t, u_l^*) \leq 2$ or $d(t, u_{l+3}^*) \leq 2$ for all $t \in \bigcup_{j=0}^{2} Y_{j+l}$. □

By Lemma 6 we have $d(v, u_{l+3}^*) \leq 2$ for all $v \in N_{l+3}$. Combining the results of Lemma 8 and Lemma 9 one can conclude the following result

Lemma 10 *All edges $(x, y) \in E$ where $x, y \in \bigcup_{j=0}^{3} N_{l+j}$ are $2NC$ by either u_l^* or u_{l+3}^* or both.*

Also from the Lemma 8 and Lemma 9 one may conclude another result which is stated below.

Corollary 1 *If $gparent(w_{l+2}) = u_l^*$ then the edge (x, y) where x, $y \in \bigcup_{j=0}^{2} N_{j+l}$ is $2NC$ by u_l^*.*

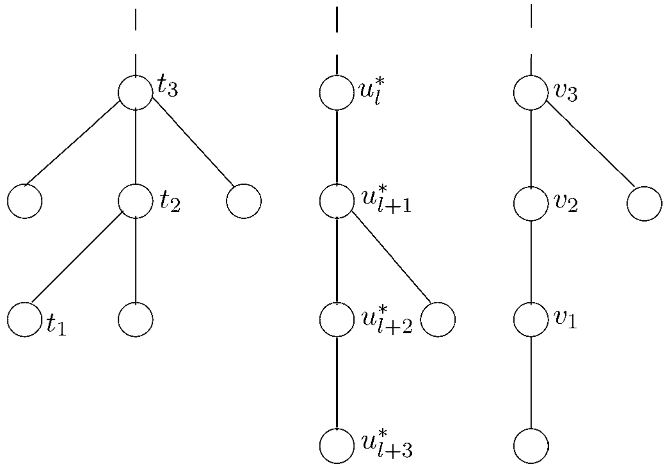

Figure 4: A part of IT

Lemma 11 *If $gparent(w_{l+2}) \neq u_l^*$ then u_{l+4}^* can not be the next member of u_l^*.*

Proof: The condition $gparent(w_{l+2}) \neq u_l^*$ implies that the IT has a branch on the left of the main path (see Figure 5).

In this case, $gparent(w_{l+2}) < u_l^*$, i.e., $parent(parent(w_{l+2})) = H(parent(w_{l+2})) < u_l^*$. So, $(parent(w_{l+2}), u_l^*) \in E$. Since, $gparent(u_{l+3}^*) < gparent(w_{l+2}) < u_l^*$ and $(gparent(u_{l+3}^*), u_l^*) \in E$ then by Lemma 1 $(gparent(w_{l+2}), u_l^*) \in E$. Therefore, the distance of the path $parent(w_{l+2}) \rightarrow gparent(w_{l+2}) \rightarrow u_l^*$ is 2. i.e., $d(parent(w_{l+2}), u_l^*) = 2$ and the distance of the path $w_{l+2} \rightarrow parent(w_{l+2}) \rightarrow gparent(w_{l+2}) \rightarrow u_l^*$ is 3. i.e., $d(w_{l+2}, u_l^*) = 3$. Thus the edge $(w_{l+2}, parent(w_{l+2}))$ is not $2NC$ by the vertex u_l^*.

Also $u_{l+3}^* < w_{l+2} < parent(u_{l+3}^*)$, so $d(w_{l+2}, u_{l+3}^*) \leq 3$. If the vertex w_{l+2} and u_{l+3}^* are adjacent then the distance of the path $parent(w_{l+2}) \rightarrow w_{l+2} \rightarrow u_{l+3}^* \rightarrow u_{l+4}^*$ is 3. Therefore the edge $(w_{l+2}, parent(w_{l+2}))$ is not $2NC$ by the vertex u_{l+4}^*. Hence u_{l+4}^* can not be the next member of u_l^*. □

Lemma 12 *If $gparent(w_{l+2}) = u_l^*$ and $X_{l+3} = \phi$ then u_{l+4}^* be a possible next member of u_l^*.*

Proof: To prove this lemma, we consider the IT of Figure 6. The relation $gparent(w_{l+2}) = u_l^*$ implies that $d(u_l^*, v) \leq 2$ for all $v \in \bigcup_{j=0}^{2} N_{j+l}$ (by Corollary 1). So the edge (x, y) where $x \in N_{l+1} \bigcup N_{l+2}$, $y \in N_{l+1} \bigcup N_{l+2}$ is $2NC$ by u_l^*.

As $X_{l+3} = \phi$, $v \leq u_{l+3}^*$, for all $v \in N_{l+3}$, i.e., $u_{l+4}^* < v < u_{l+3}^*$, for all $v \in N_{l+3}$. Again $(u_{l+3}^*, u_{l+4}^*) \in E$, so by Lemma 1, $(v, u_{l+4}^*) \in E$. Thus, $d(v, u_{l+4}^*) \leq 2$ for all $v \in N_{l+3}$. Also, $d(v, u_{l+4}^*) \leq 2$ for all $v \in N_{l+4}$. So the edge (x, y), $x \in N_{l+3} \bigcup N_{l+4}$ and $y \in N_{l+3} \bigcup N_{l+4}$ is $2NC$ by u_{l+4}^*. Hence the vertex u_{l+4}^* may be selected as the next member of u_l^*. □

From the above lemma it follows that if $X_{l+3} = \phi$ then one can select u_{l+4}^* as the next member of u_l^*. But, if $X_{l+3} \neq \phi$ then the condition for selection of u_{l+4}^* as a next member of u_l^* are discussed below.

Lemma 13 *If $gparent(w_{l+2}) = u_l^*$ and if $(u_{l+3}^*, v) \notin E$ for least one $v \in X_{l+3}$ where $X_{l+3} \neq \phi$ then u_{l+4}^* can not be the next member of u_l^*.*

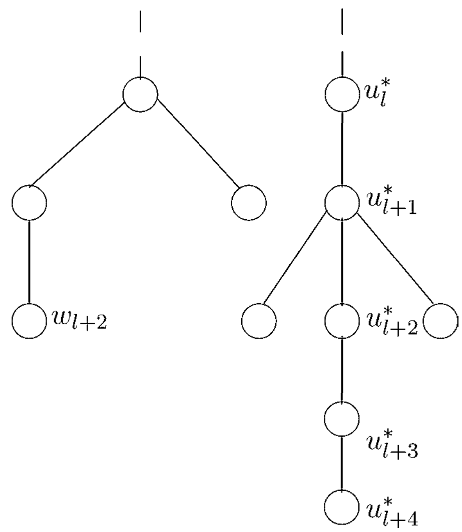

Figure 5: A part of IT

Proof: We refer the Figure 7 to prove this lemma. The relation $gparent(w_{l+2}) = u_l^*$ implies that $d(u_l^*, v) \leq 2$ for all $v \in \bigcup_{j=0}^{2} N_{j+l}$ (by Corollary 1). So all edges (x, y), $x \in N_{l+2} \bigcup N_{l+1}$ and $y \in N_{l+1} \bigcup N_{l+2}$ is $2NC$ by u_l^*. Now, $parent(u_{l+4}^*) = u_{l+3}^* = H(u_{l+4}^*)$ and $v > u_{l+3}^*$, $v \in X_{l+3}$ so $(u_{l+4}^*, v) \notin E$. Therefore, the distance of the shortest path $u_{l+4}^* \rightarrow u_{l+3}^* \rightarrow parent(v) \rightarrow v$ is 3 i.e., $d(u_{l+4}^*, v) = 3$. Thus the edge $(v, parent(v))$, $v \in X_{l+3}$ is not $2NC$ by u_{l+4}^*. Therefore, u_{l+4}^* can not be the next member of the vertex u_l^*. □

Lemma 14 *If $gparent(w_{l+2}) = u_l^*$ and if $(u_{l+3}^*, v) \in E$ for all $v \in X_{l+3}$ but $parent(v) \neq parent(u_{l+3}^*)$ for at least one $v \in X_{l+3}$ then u_{l+4}^* can not be the next member of u_l^*.*

Proof: Let v be a vertex of X_{l+3} such that $parent(v) \neq parent(u_{l+3}^*)$. In this case, distance of the path $u_l^* \rightarrow u_{l+1}^* \rightarrow parent(v)$ is 2 i.e., $d(u_l^*, parent(v)) = 2$. Also the distance of the path $u_l^* \rightarrow u_{l+1}^* \rightarrow parent(v) \rightarrow v$ is 3 i.e., $d(u_l^*, v) = 3$. So the edge $(parent(v), v)$ is not $2NC$ by u_l^* (see Figure 7).

Now, if $(u_{l+3}^*, v) \in E$ then $d(u_{l+4}^*, v) = 2$ but $H(u_{l+3}^*) = parent(u_{l+3}^*) < parent(v)$, so $(u_{l+3}^*, parent(v)) \notin E$. Hence the distance of the shortest path $u_{l+4}^* \rightarrow u_{l+3}^* \rightarrow v \rightarrow parent(v)$ is 3 i.e., $d(u_{l+4}^*, parent(v)) = 3$. Therefore the edge $(v, parent(v))$ is not $2NC$ by u_{l+4}^*. Hence u_{l+4}^* can not be the next member of u_l^*. □

Lemma 15 *If $gparent(w_{l+2}) = u_l^*$ and $(u_{l+3}^*, u) \in E$ for all $u \in X_{l+3} \bigcup Y_{l+2}$, $(v, t) \in E$ for at least one $v \in X_{l+3}$ and $t \in Y_{l+2}$ and $parent(v) = parent(u_{l+3}^*)$ for all $v \in X_{l+3}$ then u_{l+4}^* is a possible next member of u_l^*.*

Proof: Let $x \in N_{l+2} \bigcup N_{l+3}$ and $y \in N_{l+2} \bigcup N_{l+3}$. The distance of the path $u_{l+4}^* \rightarrow u_{l+3}^* \rightarrow x$ is 2 and the distance of the path $u_{l+4}^* \rightarrow u_{l+3}^* \rightarrow y$ is 2. If $(u_{l+3}^*, u) \in E$ for all $u \in X_{l+3} \bigcup Y_{l+2}$ then the edge (x, y) is $2NC$ by u_{l+4}^*. Also $d(parent(u_{l+3}^*), u_{l+4}^*) = 2$ and $d(u_{l+4}^*, v) = 2$. Also the edge $d(parent(u_{l+3}^*), v)$, $v \in X_{l+3}$ is $2NC$ by u_{l+4}^*. Again, $d(u_{l+4}^*, t) = 2$ and $d(u_{l+4}^*, t') \leq 2$, so the edge (t, t'), $t \in Y_{l+2}$, $t' \in Y_{l+3}$ is $2NC$ by u_{l+4}^*. □

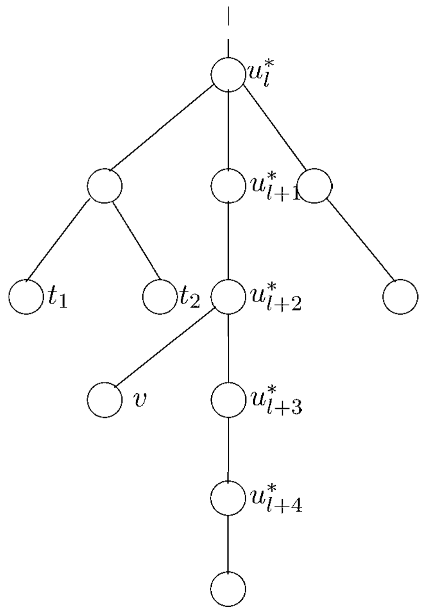

Figure 6: A part of IT

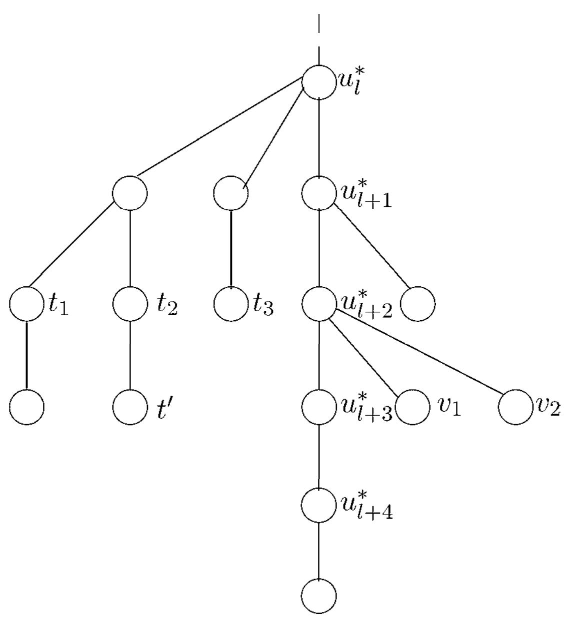

Figure 7: A part of IT

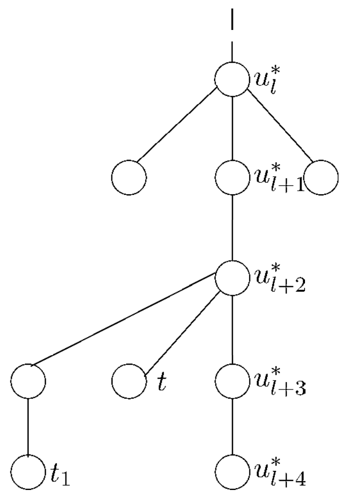

Figure 8: A part of IT

Lemma 16 *If $gparent(w_{l+2}) = u_l^*$ and $(u_{l+3}^*, v) \in E$ and $parent(v) = parenr(u_{l+3}^*)$ for all $v \in X_{l+3}$, $(v, t) \in E$ for all $v \in X_{l+3}$ and $t \in Y_{l+2}$ and $(u_{l+3}^*, t) \notin E$ for at least one $t \in Y_{l+2}$ then u_{l+4}^* can not be the next member of u_l^*.*

Proof: Since $(u_{l+3}^*, v) \in E$, $v \in X_{l+3}$, there is a path $u_{l+4}^* \to u_{l+3}^* \to v$ from the vertex u_{l+4}^* to v and hence $d(u_{l+4}^*, v) = 2$. But, the shortest path from u_{l+4}^* to t is $u_{l+4}^* \to u_{l+3}^* \to parent(u_{l+3}^*) \to t$. So, $d(u_{l+4}^*, t) = 3$. Therefore the edge (v, t), $v \in X_{l+3}$, $t \in Y_{l+2}$ is not $2NC$ by u_{l+4}^*. Also, the edge (t, v) is not $2NC$ by u_l^*. Hence, u_{l+4}^* can not be next member of u_l^*. □

Lemma 17 *If $gparent(w_{l+2}) = u_l^*$ for all $v \in X_{l+3}$, $(u_{l+3}^*, v) \in E$ and $parent(v) = parenr(u_{l+3}^*)$ and $(v, t) \notin E$ for all $v \in X_{l+3}$ and $t \in Y_{l+2}$ then u_{l+4}^* is a possible next member of u_l^*.*

Proof: We refer Figure 7 to prove this lemma. Since $(u_{l+3}^*, v) \in E$ for all $v \in X_{l+3}$ and $d(v, u_{l+4}^*) \leq 2$ then the edge (v_1, v_2), v_1, $v_2 \in X_{l+3}$ is $2NC$ by u_{l+4}^*. Let $u \in Y_{l+3}$. Since $u < u_{l+3}^*$ and $(u_{l+4}^*, u_{l+3}^*) \in E$, therefore, $(u, u_{l+3}^*) \in E$. Also, $u < u_{l+3}^* < t$, $t \in Y_{l+2}$ and if $(u, t) \in E$ and for this t, $d(u_{l+4}^*, t) = 2$. Hence (u, t) is $2NC$ by u_{l+4}^* and by Corollary 1 u_{l+4}^* may be the next member of u_l^*. □

Lemma 18 *If $X_{l+3} = \phi$ and $Y_{l+2} = \phi$ then u_{l+4}^* is a possible next member of u_l^*.*

Proof: For this case, a possible IT is shown in the Figure 8. Let $t \in Y_{l+3}$ and $t_1 \in Y_{l+4}$. As $u_{l+4}^* < t < u_{l+3}^*$ and $(u_{l+4}^*, u_{l+3}^*) \in E$ then $(t, u_{l+3}^*) \in E$ and hence $d(t, u_{l+4}^*) \leq 2$.

Also $d(t_1, u_{l+4}^*) \leq 2$ (by the Lemma 6). Thus the edge (t, t_1), if any, is $2NC$ by u_{l+4}^*. And by Corollary 1, the lemma follows. □

Lemma 19 *If $Y_{l+2} = \phi$ and $(u_{l+3}^*, v) \notin E$ for at least one $v \in X_{l+3}$ then u_{l+4}^* can not be the possible next member of u_l^*.*

Proof: We refer Figure 9 to prove this lemma. If $(u_{l+3}^*, v) \notin E$ for at least one $v \in X_{l+3}$ then the shortest path from u_{l+4}^* to v is $u_{l+4}^* \to u_{l+3}^* \to parent(u_{l+3}^*) \to v$. Therefore, $d(u_{l+4}^*, v) = 3$. Hence the edge (u, v), $u \in X_{l+2}$, $v \in X_{l+3}$ is not $2NC$ by u_{l+4}^*. Thus u_{l+4}^* can not be the possible next member of u_l^*. □

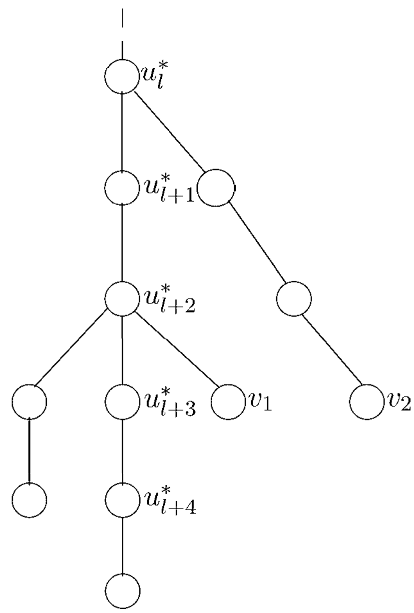

Figure 9: A part of IT

Lemma 20 *If $Y_{l+2} = \phi$ and $(u^*_{l+3}, v) \in E$ for all $v \in X_{l+3}$ and $parent(v) \neq parent(u^*_{l+3})$ for at least one $v \in X_{l+3}$ then u^*_{l+4} can not be the possible next member of u^*_l.*

Proof: Without loss of generality, we assume that $(u^*_{l+3}, v_2) \in E$ and $parent(v_2) \neq parent(u^*_{l+3})$, $v_2 \in X_{l+3}$ (see Figure 9). Since, $parent(v_2) \neq parent(u^*_{l+3})$, $(u^*_{l+3}, parent(v_2)) \notin E$ as $parent(u^*_{l+3})$ $= H(u^*_{l+3}) < parent(v_2)$. In this case, the path from u^*_{l+4} to $parent(v)$ is $u^*_{l+4} \to u^*_{l+3} \to v_2 \to parent(v_2)$.

Therefore, $d(u^*_{l+4}, parent(v_2)) = 3$ and $d(u^*_{l+4}, v_2) = 2$. Hence the edge $(v_2, parent(v_2))$ is not $2NC$ by the vertex u^*_{l+4}. Thus, u^*_{l+4} can not be the possible next member of u^*_l. □

Lemma 21 *If $Y_{l+2} = \phi$ and $d(u^*_{l+3}, v) \in E$ for all $v \in X_{l+3}$ and $parent(v) = parent(u^*_{l+3})$ for all $v \in X_{l+3}$ then u^*_{l+4} may be the possible next member of u^*_l.*

Proof: For this case, a possible IT is shown in Figure 10. Since, $d(u^*_{l+3}, v) \in E$ for all $v \in X_{l+3}$ and $d(u^*_{l+4}, v) = 2$ as $u^*_{l+4} \to u^*_{l+3} \to v$. Also, $d(u^*_{l+4}, t) \leq 2$ for all $t \in Y_{l+3}$. Again, $Y_{l+2} = \phi$ and $parent(v) = parent(u^*_{l+3})$ for all $v \in X_{l+3}$, so the edge $(parent(u^*_{l+3}), u)$, $u \in N_{l+3}$ is $2NC$ by u^*_{l+4}.

If $d(u^*_{l+4}, u_1) \leq 2$ and $d(u^*_{l+4}, v) = 2$ where $v \in X_{l+3}$, $u_1 \in X_{l+4}$, then the edge (v, u_1) is also $2NC$ by the vertex u^*_{l+4}. Hence, u^*_{l+4} may be the possible next member of u^*_l. □

Lemma 22 *The set C is 2-neighbourhood covering set of the circular-arc graph G, where $C = C^*_i$ for $i = 1, 2, 3, 4$ and $|C^*_i|$ =min $\{|C^*_1|, |C^*_2|, |C^*_3|, |C^*_4|\}$.*

Proof: Let u^*_1 is the first vertex of the set C^*_1. The next vertex of the C^*_1 is either u^*_{l+3} or u^*_{l+4}. In C^*_1, let the distance between two consecutive vertices be 4 i.e., the vertices of the set C^*_1 are u^*_1, u^*_5, u^*_9, u^*_{13} etc. Also in C^*_4 let the distance between two consecutive vertices be 3 i.e., the vertices of the set C^*_4 are u^*_4, u^*_7, u^*_{10}, u^*_{13} etc. Now, u^*_{13} is a member in both sets the C^*_1 and

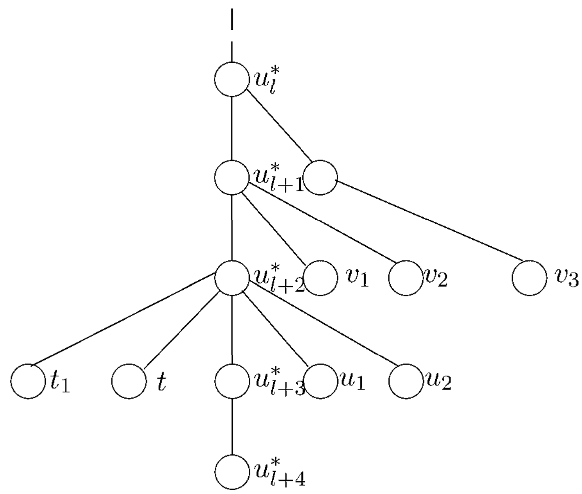

Figure 10: A part of IT

C_4^*. From the above lemmas we observed that after the vertex u_{13}^*, both the sets C_1^* and C_4^* are same. Let u_k^* be the last vertex of the sets C_1^* and C_4^* and let $u_k^* = parent(u_1^*)$. The edge (u_1^*, u_2^*) is $2NC$ by the vertex u_k^* and the edge (u_2^*, u_3^*) is $2NC$ by the vertex u_4^*. So, there are same number of vertices in both the sets C_1^* and C_4^*. Therefore, in this case the cardinality of C_1^* and C_4^* are same.

Let there exist a vertex v at level 3 adjacent with only u_2^*. Then the distance of the path $v \to u_2^* \to u_1^* \to u_k^*$ is 3. Also the distance of the path $v \to u_2^* \to u_3^* \to u_4^*$ is 3. So, the edge (v, u_2^*) is not $2NC$ by neither the vertex u_k^* nor the vertex u_4^*. For the edge (v, u_2^*) any one of the vertices u_1^*, u_2^*, u_3^* must be a member of the set C_4^*. In that case, the number of vertices of the set C_1^* is less than the number of vertices of the set C_4^*, before u_{13}^*. So, the cardinality of C_4^* is greater than the cardinality of the set C_1^*. Similarly, the cardinalities of C_1^*, C_2^*, C_3^*, C_4^* may or may not be equal.

In C_5^*, let the distance between two consecutive vertices be 4 i.e., the vertices of the set C_5^* are u_5^*, u_9^*, u_{13}^* etc. Also in C_3^* let the distance between two consecutive vertices be 3 i.e., the vertices of the set C_3^* are u_3^*, u_6^*, u_9^* etc. We know, u_5^* is the first vertex of set C_5^*. The distance of the path $u_3^* \to u_4^* \to u_5^*$ is 2 and also the distance of the path $u_2^* \to u_3^* \to u_4^* \to u_5^*$ is 3. Then the edge (u_2^*, u_3^*) is not $2NC$ by the vertex u_5^*. Similarly, the distance of the path $u_2^* \to u_1^* \to parent(u_1^*)$ is 2 and the distance of the path $u_3^* \to u_2^* \to u_1^* \to parent(u_1^*)$ is 3. So, the edge (u_2^*, u_3^*) is not $2NC$ by the vertex $parent(u_1^*)$. For the edge (u_2^*, u_3^*) any one of the vertices u_1^*, u_2^*, u_3^*, u_4^* is a member of C_5^*. Let u_3^* be the terminal vertex of the set C_5^*. So, from the vertex u_9^* to the vertex there are same vertices in both the sets C_3^* and C_5^*. Also, there are two vertices u_3^*, u_6^* in C_3^* before u_9^* and there are two vertices u_3^*, u_5^* in C_5^* before u_9^*. So the cardinality of the set C_3^* is same as the cardinality of the set C_5^*.

Similarly, the cardinality of the set C_i^* where $i = 5, 6, \ldots h'$ is same to any one of the cardinality of the sets C_1^*, C_2^*, C_3^*, C_4^*. Therefore the set C is the 2-neighbourhood covering set of the circular-arc graph G. □

6 Algorithm and its Complexity

In this section, we present an algorithm to find the 2-neighbourhood covering set of a circular-arc graph. The time complexity is also calculated here.

Here a procedure **FINDNEXTC** is formally presented in the following which computes the level L of next vertex u_L^* of C_i^*, if the level l of the current vertex u_l^* is supplied.

Procedure FINDNEXTC(l, L)

// This procedure computes the level L such that u_L^* will be the next number of C_i^* where as u_l^* is the currently selected vertex of C_i^*. The sets X_j, Y_j and the array $u_j^*, j = 1, 2, \ldots, h$, h is the height of the tree $T_i^*(G)$ are known globally. //

```
Initially L = l + 3
If Y_{l+2} = φ then
    if X_{l+3} = φ then L = l + 4 (Lemma 18)
    elseif for all v ∈ X_{l+3}, parent(v) = parent(u*_{l+3}) and (u*_{l+3}, v) ∈ E then
        L = l + 4; (Lemma 21)
    endif ;
else // Y_{l+2} ≠ φ //
    if gparent(w_{l+2}) = u*_l then
        if X_{l+3} = φ then L = l + 4; (Lemma12)
        elseif for all v ∈ X_{l+3}, parent(v) = parent(u*_{l+3}),  (u*_{l+3}, v) ∈ E and
            if (v, t) ∈ E for some v ∈ X_{l+3}, t ∈ Y_{l+2} and
                (u*_{l+3}, t) ∈ E then L = l + 4; (Lemma 15)
            elseif (v, t) ∉ E for all v ∈ X_{l+3} and t ∈ Y_{l+2}
                then L = l + 4; (Lemma 17)
            endif ;
        endif;
    endif;
endif;
return L;
end FINDNEXTC
```

Now we present an algorithm to find C_i^*, for all $i = 1, 2, 3, 4$ from the interval trees T_i^*, for $i = 1, 2, 3, 4$.

Algorithm FOURTNC $(T_i^*(G))$
Input: An interval tree $T_i^*(G)$ and the vertex u_i^*, $i \in \{1, 2, 3, 4\}$.
Output: The 2-neighbourhood-covering set C_i^*.
Initially, $C_i^* = \phi$ (null set), $l = 0$.
Step 1: Construct the interval tree $T_i^*(G)$.
Step 2: Compute the vertices on the main path of the tree $T_i^*(G)$ and let them $v_j^*, j = 1, 2, \ldots, h$, h is the height of the tree $T_i^*(G)$
Step 3: Compute the set of X_j and Y_j, for each $j = 1, 2, \ldots, h$.
Step 4: $C_i^* = C_i^* \bigcup \{v_0^*\}$. Set $p_i^* = n - u_i^*$.
Step 5: Repeat
Call **FINDNEXTC**(l, L); // Find level L for the next vertex of C_i^*. //
$l = L$;
if $v_l^* > p_i^*$ then $v_l^* = v_l^* - p_i^*$.

if $v_l^* \le p_i^*$ then $v_l^* = v_l^* + (n - p_i^*)$.
$C_i^* = C_i^* \bigcup \{v_l^*\}$.
Until $\{|h - l| \le 3\}$.
end FOURTNC

After finding four sets $C_1^*,\ C_2^*,\ C_3^*,\ C_4^*$, the complete algorithm to find 2-neighbourhood covering set is given below.

Algorithm CTWONC
Input: A family of circular arcs A of a circular-arc graph G.
Output: Minimum cardinality 2-neighbourhood-covering set C.
Step 1: Construct the interval tree $T(G)$ rooted at n.
Step 2: Compute the vertices on the main path of the tree $T(G)$ and let them be $u_i^*,\ i = 1, 2, \ldots, h'$, h' is the height of the tree $T(G)$.
Step 3: Construct the four interval trees $T_1^*(G),\ T_2^*(G),\ T_3^*(G),\ T_4^*(G)$, where $u_1^*,\ u_2^*,\ u_3^*,\ u_4^*$. are respective roots.
Step 4: Compute four $2NC$ sets $C_1^*,\ C_2^*,\ C_3^*,\ C_4^*$ by Algorithm **FOURTNC**.
Step 5: Set $C = C_i^*$, where $|C_i^*|$ =min $\{|C_1^*|,\ |C_2^*|,\ |C_3^*|,\ |C_4^*|\}$.
end CTWONC

The vertices of $T_i^*(G)$ are the vertices of G. The sets $N_j,\ j = 1, 2, \ldots h$ are mutually exclusive and the vertices of each N_j are consecutive integers. Again, the sets X_j and $Y_j,\ j = 1, 2, \ldots, h$ are also mutually exclusive, i.e., $X_j \bigcap X_k = \phi,\ Y_j \bigcap Y_k = \phi$, for $j \ne k$ and $j,\ k = 1, 2, \ldots, h$ and $X_j \bigcap Y_k = \phi,\ j,\ k = 1, 2, \ldots, h$. Moreover, $N_j = X_j \bigcup Y_j \bigcup \{u_j^*\},\ j = 1, 2, \ldots h$. The vertices of each X_j and Y_j are also consecutive integers. So only the lowest and highest numbered vertices are sufficient to maintain the sets $X_j,\ Y_j,\ N_j,\ j = 1, 2, \ldots, h$. So, we will store only lowest and highest numbered vertices corresponding the sets $X_j,\ Y_j,\ N_j$ instead of all vertices. If any set is empty then the lowest and highest numbered vertices may be taken as 0 and 0. It is obvious that $|\bigcup_{j=0}^{n} N_j| = n + 1$. In the procedure **FINDNEXTC**, only the vertices of the set $N_l,\ N_{l+1},\ N_{l+2}$ and N_{l+3} are considered to process then the total number of vertices of these sets is $|\bigcup_{j=0}^{3} N_{j+l}|$ and the subgraph induced by the vertices $|\bigcup_{j=0}^{3} N_{j+l}|$ is a part of the tree $T_i^*(G)$. So the total number of edges in this portion is less than or equal to $|\bigcup_{j=0}^{3} N_{j+l}|$. Hence one can conclude the following result.

Lemma 23 *The time complexity of the procedure* **FINDNEXTC**(l, L) *is* $O(|\bigcup_{j=0}^{3} N_{j+l}|)$.

In the following we compute the time complexity of Algorithm **FOURTNC**.

Theorem 1 *The 2-neighbourhood covering set of any one of the interval graphs G_i^* can be computed in $O(n)$ time.*

Proof: For a given interval representation of an interval graph, the interval tree $T_i^*(G)$ can be constructed in $O(n)$ time [10, 11]. Since the main path starting from the vertex 0 and ending at the vertex n, all the vertices $u_j^*,\ j = 1, 2, \ldots, h$ on the main path can be identified in $O(n)$ time. By computing the level of each vertex one can compute the sets X_i and $Y_i,\ j = 1, 2, \ldots, h$ in $O(n)$ time. Step 3 of Algorithm **FOURTNC** can be computed in $O(n)$ time. Each iteration of repeat-until loop takes only $O(|\bigcup_{j=0}^{3} N_{j+l}|)$ time for a given l. The Algorithm **FOURTNC** calls the procedure **FINDNEXTC** for $|C_i^*|$ time and each time the value of l is increased by either 3 or 4. Also, if the vertices of the set $\bigcup_{j=0}^{3} N_{j+l}$ or $(\bigcup_{j=0}^{3} N_{j+l'})$ are consider to find the

kth (k'th) member of C then $\bigcup_{j=0}^{3} N_{j+l}$ and $\bigcup_{j=0}^{3} N_{j+l'}$ are disjoint. Therefore, Step 5 takes $O(|\bigcup_{j=0}^{h} N_j|) = O(n)$ time. Hence the time complexity of the Algorithm **FOURTNC** is $O(n)$. □

Lemma 24 *The time complexity of Algorithm* **CTWONC** *is* $O(n)$.

Proof: For a given interval graph representation, the unique interval tree $T(G)$ can be constructed in $O(n)$ time. So, in algorithm **CTWONC**, Step 1 takes $O(n)$ time. The vertices of the main path of the tree $T(G)$ can be identified in $O(n)$ time. So, the Step 2 take $O(n)$ time. Also Step 3 takes $O(n)$. For each interval tree $T_i^*(G)$ the 2-NC set C_i^* can be computed in $O(n)$ time (Lemma 23). So, Step 4 takes $O(n)$ time. Step 5 easily can be computed in $O(n)$ time. Hence the overall time complexity of Algorithm **CTWONC** is $O(n)$. □

References

[1] Change, G. J., Farber, M. and Tuza, Z. (1993). Algorithmic aspectd of neighbourhood numbers, *SIAM J. Discrete Math.*, **6**, 24-29.

[2] Corneil, D. G., Olariu, S. and Stewart, L. (1998). The ultimate interval graph recognition algorithm, *In Proceding of the ninth Annual ACM-Symposium on Discrete Algorithms (SODA)*, 175-180.

[3] Eschen, E. M. and Spinrad, J. P. (1993). An $O(n^2)$ algorithm for circular-arc graph recognition, *In Proc: 4th Annual ACM-SIAM Symposium on Discrete Algorithm, Austin, TX*, 128-137.

[4] M. C. Golumbic, *Algorithm Graph Theory and Perfect Graphs*, (2004). Elsevier.

[5] Hsu W. -L. and Tsai, K. -H. (1991). Linear time algorithms on circular-arc graphs, *Information Processing Letters*, **40**, 123-129.

[6] Hwang, S. F. and Chang, G.J. (1998). k-neighbourhood-covering and independence problems for chordal graphs, *SIAM J. Discrete Math.*, **11(4)**, 633-643.

[7] Lehel, J. and Tuza, Z. (1986). Neighbourhood perfect grsphs, *Discrete Math.*, **61**, 93-101.

[8] Mondal, S., Pal, M. and Pal, T. K. (2002). An optimal algorithm to solve 2-neighbourhood covering problem on interval grsphs, *Intern. J. Computer Math.*, **79(2)**, 189-204.

[9] Pal, M. and Bhattacharjee, G. P. (1996). A sequential algorithm for finding a maximum weight k-independent set on interval graphs, *Intern. J. Computer Math.*, **60**, 205-214.

[10] Pal, M. and Bhattacharjee, G. P. (1997). A data structure on interval graphs and its applications, *J. Circuits, Systems and Computers*, **7(3)** 165-175.

[11] Pal, M. and Bhattacharjee, G. P. (1997). An optimal parallel algorithm for all-pairs shortest paths on unweighted interval graphs, *Nordic J. Computing*, **4**, 342-356.

[12] Saha, A., Pal, M. and Pal, T. K. (2005). An optimal parallel algorithm to construct a tree 3-spanner on interval graphs, *Intern. J. Computers Math.*, **82**, 259-274.

[13] Saha, A., Pal, M. and Pal, T. K. (2005). An optimal parallel algorithm for solving all-pairs shortest paths problem on circular-arc graph, *J. Applied Mathematics and Computing*, **17**, 1-23.

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