AMO - Advanced Modeling and Optimization, Volume 9, Number 1, 2007

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

Prabir K. Ghosh and Madhumangal Pal

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

Abstract

Let $G = (V, E)$ be a simple graph and k be a fixed integer. A vertex z is said to be a k-neighbourhood-cover of an edge (x, y) if $d(x, z) \leq k$ and $d(y, z) \leq k$, where $d(x, y)$ represents the distance between two vertices x and y. A set $C \subset V$ is called a k-neighbourhood-covering set if every edge in E is k- neighbourhood-cover by some vertices of C. This problem is NP-complete for general graphs even it remains NP-complete for chordal graphs. Using dynamic programming technique, an $O(n)$ time algorithm is designed to solve minimum 2-neighbourhood-covering problem on trapezoid graph. The trapezoid interval tree rooted at the vertex n is used to solve this problem.

Keywords: Design and analysis of algorithms, tree, 2-neighbourhood-covering, trapezoid graph.

1 Introduction

1.1 Trapezoid graph

A trapezoid i is defined by four corner points $[a_i, b_i, c_i, d_i]$, where $a_i < b_i$ and $c_i < d_i$ with a_i, b_i lying on the top channel and c_i, d_i lying on the bottom channel of the trapezoid diagram. An undirected graph $G = (V, E)$ is called a *trapezoid graph* if it can be represented by a trapezoid diagram such that each vertex v_i in V corresponds to a trapezoid i and $(v_i, v_j) \in E$ if and only if the trapezoids i and j corresponding to the vertices v_i and v_j intersect in the trapezoid diagram.

Figure 1 represent a trapezoid graph and its corresponding trapezoid diagram. The class of trapezoid graphs includes two well known classes of intersection graphs: the permutation graphs and the interval graphs [4]. The permutation graphs are obtained in the case where $a_i = b_i$ and $c_i = d_i$ for all i, and the interval graphs are obtained in the case where $a_i = c_i$ and $b_i = d_i$ for all i. Let $T = \{1, 2, \ldots, n\}$, be the n trapezoids where trapezoid i is represented in the trapezoid diagram by four corner points $[a_i, b_i, c_i, d_i]$, a_i, c_i being the left corner points and b_i, d_i being the right corner points. Without any loss of generality we assume the following:

(a) a trapezoid contains four different corner points and that no two trapezoids share a common end point,

(b) trapezoids in the trapezoid diagram and vertices in the trapezoid graph are one and same thing,

(c) the trapezoids in the trapezoid diagram T are indexed by increasing right end points on the top channel i.e., $1 < 2 < \cdots < n$ if and only if $b_1 < b_2 < \cdots < b_n$.

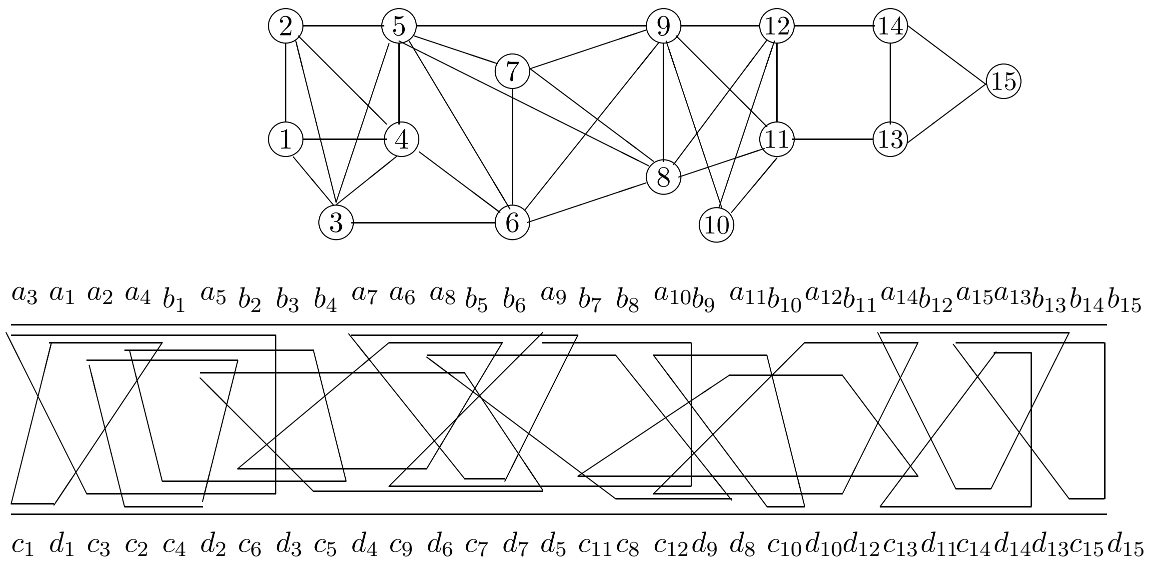

Figure 1: A trapezoid graph G and its trapezoid representation.

1.2 The k-neighbourhood-covering 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. 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 vertex x to vertex y is the minimum length of a path from x to y, and if there is no path from x to y then $d(x, y)$ is taken as ∞.

A vertex x k-dominates another vertex y if $d(x, y) \leq k$. A vertex z 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 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 vertices in C. The k-NC number $\rho(G, k)$ of G is the minimum cardinality of all k-NC sets.

1.3 Review of previous works

Lehel et al. [3] have presented a linear time algorithm for computing the k-NC number $\rho(G, k)$ for $k = 1$, *i.e.*, $\rho(G, 1)$ for an interval graph. Chang et al. [2] and Hwang et al. [9] have presented linear time algorithms for computing $\rho(G, 1)$ for a strongly chordal graph provided that strong elimination ordering is known. Hwang et al. [9] also proved that k-NC problem is NP-complete for chordal graphs. Mondal et al. [10] have presented a linear time algorithm for computing 2-NC problem for an interval graph.

1.4 Our result

To find the 2-neighbourhood-covering (2-NC) set, we construct a trapezoid interval tree (TIT) rooted at the vertex n. The TIT is computed in $O(n)$ time. Based on this TIT, we design an algorithm to find the minimum 2-NC set of the trapezoid graph, using dynamic programming technique. The proposed algorithm takes $O(n)$ time and $O(n)$ space.

2 Preliminaries

Let $G = (V, E)$ be a trapezoid graph, where $V = \{1, 2, \ldots, n\}$ be the set of vertices of G. We define some terms which are necessary to solve this problem.

Definition 1 Right spread. *The right spread of a trapezoid T_i is the maximum of b_i and d_i,i.e., right spread of a trapezoid T_i or the vertex i is $max\{b_i, d_i\}$.*

An array $f(i)$ is defined as follows:

$$f(i) = max\{b_i, d_i\}, i \in V.$$

That is, the array $f(i)$ is the right spread of all the vertices $i \in V$.

vertex i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a_i	2	3	1	4	6	11	10	12	15	18	20	22	27	24	26
b_i	5	7	8	9	13	14	16	17	19	21	23	25	28	29	30
c_i	1	4	3	5	9	7	13	17	11	21	16	18	24	26	29
d_i	2	6	8	10	15	12	14	20	19	22	25	23	28	27	30
$f(i)$	5	7	8	10	15	14	16	20	19	22	25	25	28	29	30

Table 1: The arrays a_i, b_i, c_i, d_i and $f(i)$.

Now, to find the minimum 2-NC set, we rearranged the vertex set V according to the increasing order of $f(i)$, for all $i \in V$. Let this arranged vertex set be V'. This means that if $f(i) < f(j)$ in V then $i < j$ in V'. In fact, V is renamed as V'. We rename the trapezoid graph G as G' where $G' = (V', E')$, $E' = \{(u, v) \in E \; \forall u, v \in V'\}$. It is obviously that $|V| = |V'| = n$, where n is the number of vertices of V. Figure 2 represents the trapezoid graph G'.

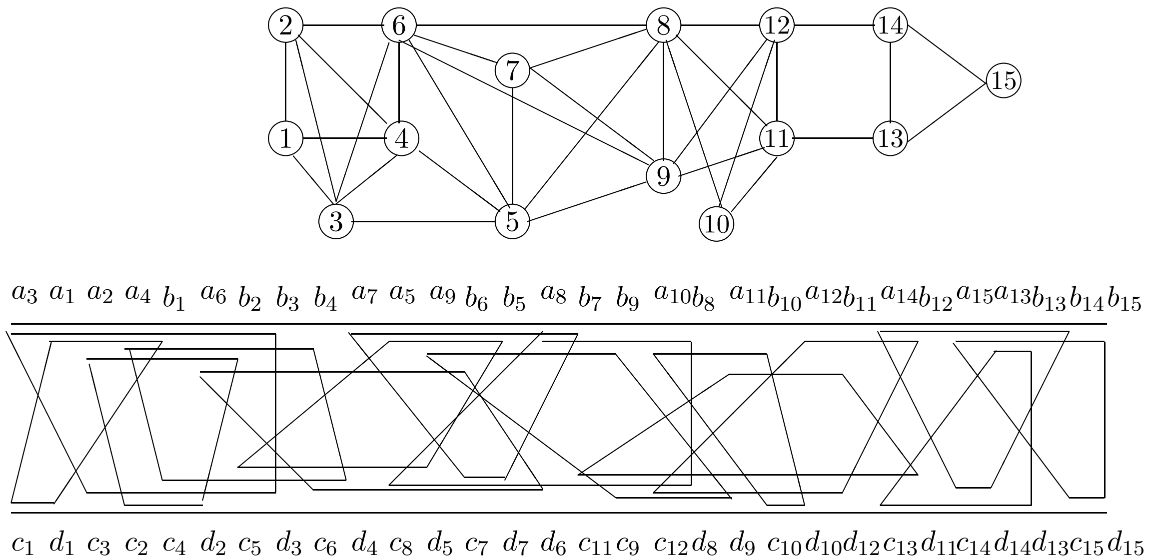

Figure 2: A trapezoid graph G' and its trapezoid representation.

The arrays a_i, b_i, c_i, d_i and $f(i)$ of the graph of Figure 1 are shown in Table 1.

2.1 Interval representation of a trapezoid graph

Let $I' = \{I'_1, I'_2, \ldots, I'_n\}$, $I'_j = [p_j, q_j]$, $p_j = min\{a_j, c_j\}$ and $q_j = max\{b_j, d_j\}$, $j = 1, 2, \ldots, n$, be the interval representation of the trapezoid graph $G' = (V', E')$. p_j and q_j respectively called the left and right endpoints of the interval I'_j. Without loss of generality, we assume that each interval contains both of its end points and that no two intervals share a common endpoints. If the intervals have common endpoints then the algorithm CONVERT [6] may be used to convert the intervals of I' into intervals of distinct endpoints. We consider intervals in the set I' rather then the vertices in G'. Further the trapezoid graph G is connected. Therefore G' is also connected.

Definition 2 Parallel trapezoids. *Two trapezoids T_i and T_j of a trapezoid graph are parallel if their corresponding intervals I_i and I_j have a common line segment or a common point but the trapezoids T_i and T_j are not intersect.*

It is interesting that if two trapezoids, say, T_i and T_j are parallel of a trapezoid graph G' then their corresponding intervals, say, I'_i and I'_j have a common line segment or a common point. Let the sorted endpoints are available and the intervals in I' are indexed by increasing right endpoints *i.e.*, $q_1 < q_2 < \cdots < q_n$. This indexing is known as interval ordering of the corresponding trapezoid graph G'. This ordering is unique when a representation by a set of intervals is provided and fixed. The interval representation of the trapezoid graph G' of Figure 2 is shown in Figure 3.

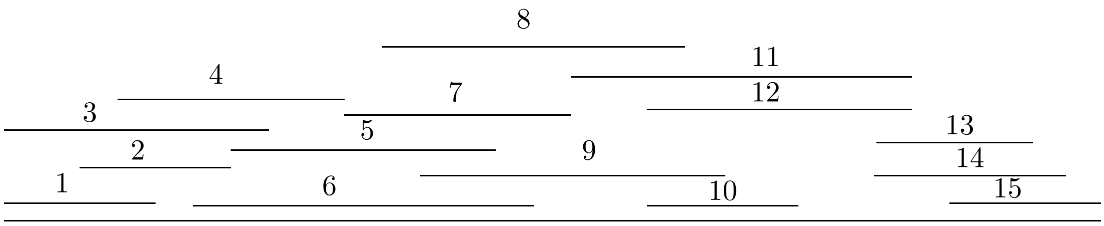

Figure 3: An interval representation of G'.

2.2 Some results on trapezoid graph

In this section, we present some important results of a trapezoid graph those are necessary to develop the algorithm to find 2-neighbourhood-covering of trapezoid graph.

Lemma 1 *[7] If the vertices $u, v, w \in V$ are such that $u < v < w$ and u is adjacent to w, then either v is adjacent to u or v is adjacent to w.*

In a trapezoid diagram, two trapezoids T_i and T_j are not adjacent if the trapezoids T_i and T_j satisfied Lemma 2.

Lemma 2 *[1] Two vertices i and j of a trapezoid graph are not adjacent iff either (i) $b_i < a_j$ and $d_i < c_j$ or (ii) $b_j < a_i$ and $d_j < c_i$.*

In a trapezoid diagram, two trapezoids T_i and T_j are parallel if the trapezoids T_i and T_j satisfy the following result.

Lemma 3 *For two trapezoids T_i and T_j, if $b_i < a_j$ and $d_i < c_j$ then T_i and T_j are parallel iff $b_i < a_j \leq d_i$ or $d_i < c_j \leq b_i$, for $i < j$.*

Proof. To prove this lemma, refer Figure 4.

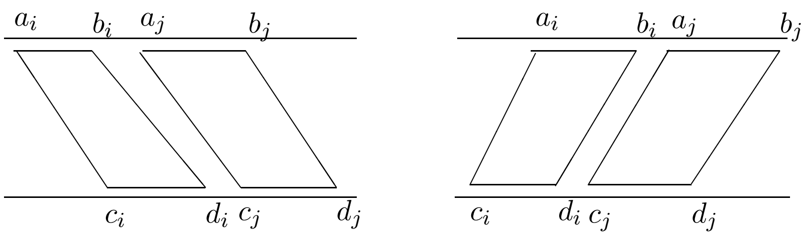

Figure 4: Two types of parallel trapezoids.

Let i and j be two vertices of a trapezoid graph corresponding to the trapezoids T_i and T_j respectively. If $b_i < a_j$ and $d_i < c_j$ then in trapezoid diagram, the trapezoids T_i and T_j have no common region *i.e.*, $(i, j) \notin E$. Let $b_i < a_j \leq d_i$ or $d_i < c_j \leq b_i$ for $i < j$. This means that the reduce intervals of the corresponding trapezoids T_i and T_j of a trapezoid graph have a common line segment or a common point, implying that the trapezoids T_i and T_j are parallel. Conversely, if $b_i < a_j$ and $d_i < c_j$ *i.e.*, $(i, j) \notin E$ then the trapezoids T_i and T_j are parallel only when the reduced intervals of a trapezoid graph have a common line segment or a common point, *i.e.*, $b_i < a_j \leq d_i$ or $d_i < c_j \leq b_i$ for $i < j$. □

From the graph of Figure 1, the trapezoid T_2 is parallel to T_6, T_4 is parallel to T_7, T_7 is parallel to T_{11}, T_8 is parallel to T_{10}, T_{11} is parallel to T_{14} and T_{12} is parallel to T_{13}.

Therefore, in the graph of Figure 2, the trapezoid T_2 is parallel to T_5, T_4 is parallel to T_7, T_7 is parallel to T_{11}, T_9 is parallel to T_{10}, T_{11} is parallel to T_{14} and T_{12} is parallel to T_{13}.

Let $H(x)$ be the highest numbered adjacent vertex of x for each $x \in V'$. If there is no vertex adjacent to x and greater then x then $H(x)$ is assumed to be x. In other words,

vertex i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$H(i)$	4	6	6	6	9	9	9	12	12	12	13	14	15	15	15

Table 2: The vertices i and the array $H(i)$ for the graph of Figure 3.

$$H(x) = max\{y : (y, x) \in E', y \geq x, \ x, y \in V'\}.$$

The array $H(x)$, $x \in V'$ satisfied the following result.

Lemma 4 *[5] If $x, y \in V'$ and $x < y$ then $H(x) \leq H(y)$.*

For the graph of Figure 3, the vertex i and the array $H(i)$ are shown in Table 2.

Now, we define TIT $T(G')$ rooted at n for a trapezoid graph G' as $T(G') = (V', E'')$ where $E'' = \{(x, y) : x \in V'$ and $y = H(x), x \neq n\}$.

The TIT $T(G')$ of the interval representation of Figure 3 is shown in Figure 5.

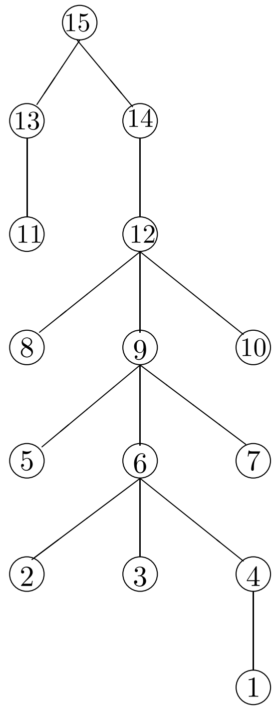

Figure 5: TIT of the trapezoid graph of Figure 2 .

The children and parent of the vertices of $T(G')$ are shown in Table 3.

Since the tree $T(G')$ is built from the vertex set V' and the edge set $E'' \subseteq E'$. Let N_j be the set of vertices which are at a distance j from the vertex n in TIT. Thus

Parent	Children
4	1
6	2, 3, 4
9	5, 6, 7
12	8, 9, 10
13	11
14	12
15	13, 14

Table 3: Parent and children of the tree of Figure 5.

$N_j = \{u : d(u, n) = j\}$ and N_0 is the singleton set $\{n\}$.

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

The property that the vertices at any level of TIT are the consecutive integers, is proved in [5] which is stated below.

Lemma 5 *[5] The vertices of N_j are consecutive integers and if x is equal to $min\{u : u \in N_j\}$ then $max\{u : u \in N_{j+1}\}$ is equal to $x - 1$.*

The following result is also proved in [5].

Lemma 6 *If $level(x) < level(y)$ then $x > y$.*

If the level of a vertex x of TIT is j then it should be adjacent to the vertices at levels $j-1, j$ and $j+1$ in G'. This observation is proved in the following lemma.

Lemma 7 *[8] If u and v be any two vertices of TIT and if $|level(u) - level(v)| > 1$ then (u, v) does not belong to E as well as E'.*

The distance $d(u, v)$ between the vertices u and v of same level and same parent is either 1 or 2, is given by in the following lemma.

Lemma 8 *[8] For $u, v \in V$ if level(u) = level(v) and parent(u) = parent(v) then distance between u and v in G is given by*

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

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

The path in TIT from the vertex 1 to the root n is called *main path*. We denote the vertex at level l on the main path by u_l^* for all l. It is obvious that *level(1)* is equal to the height (h) of the tree TIT.

3 2-Neighbourhood-Covering set

Let C be the minimum 2-NC set of the given trapezoid graph G. Therefore C is also the minimum 2-NC set of the given trapezoid graph G'. To find a 2-NC set on trapezoid graphs, a TIT is to be constructed.

The basic idea to compute 2-NC is described below. If there exists at least one vertex of N_1 which is not adjacent to u_1^*, we take u_1^* as a member of C otherwise we select the vertex u_2^* as a member of C. Let the first selected vertex (u_1^* or u_2^*) be at level l. After selection of first member of C, we are consider two vertices u_{l+2}^* and u_{l+3}^* on the main path at level $l+2$ and $l+3$ respectively. Now either u_{l+2}^* or u_{l+3}^* (not both) will be a member of C. This selection is to be made according to same results, discussed in the following. After selection of second member of C, we set $l+2$ to l, if u_{l+2}^* is selected, otherwise we set $l+3$ to l. This selection is to be continued till new $l+2$ becomes greater than the height of the tree TIT.

3.1 Selection of first member of C

The condition to select u_1^* as a first member of C is obtained in the following lemma.

Lemma 9 *If there exists at least one vertex of N_1 which is not connected with u_1^*, then u_1^* is a possible member of C.*

Proof. From the tree TIT it is clear that n is the parent of u_1^*. Let there exist at least one vertex at level 1, *i.e.*, in N_1 which is not connected with u_1^*. Let v_1 be any such vertex. Then $d(u_1^*, v_1) = 2$ (as $u_1^* \rightarrow n \rightarrow v_1$) and $d(u_1^*, n) = 1$, *i.e.*, the vertex u_1^* is a 2-NC of the edge (v_1, n). If v_2 be any vertex of N_1 connected with u_1^* then $d(v_2, n) = 1$. As $d(n, u_1^*) = 1$, u_1^* is also a 2-NC of the edge (v_2, n). Hence u_1^* is a 2-NC of (v_1, n) for each $v_1 \in N_1$. □

If u_1^* is connected with all vertices of N_1 then for all $v \in N_1$, $d(v, u_1^*) = 1$. In this case, the vertex u_2^* is to be selected as a member of C. This result is proved in the following lemma.

Lemma 10 *If u_1^* is connected with all vertices of N_1 then u_2^* is a possible member of C.*

Proof. Let u_1^* be connected with all vertices of N_1. Therefore, $d(u_1^*, v) = 1 = d(u_1^*, n)$ for all $v \in N_1$. Hence the path from u_2^* to any vertex v, $v \in N_1$ is $u_2^* \to u_1^* \to v$ (Since u_1^* is adjacent with all vertices of N_1), so $d(u_2^*, v) = 2$. But u_2^* may be adjacent to some vertices of N_1. In this case $d(u_2^*, v) = 1$. Hence $d(u_2^*, v) \le 2$, for all $v \in N_1$. Also, $d(u_2^*, n) = 2$. Thus, the edge (n, v), $v \in N_1$ are 2-NC by u_2^*.

Again, if $v' \in N_2$ then $d(u_2^*, v') \le 2$ (Lemma 1). Therefore, $d(u_2^*, v) \le 2$ and $d(u_2^*, v') \le 2$ for $v \in N_1$ and $v' \in N_2$. Thus each edge $(v, v') \in E'$ is 2-NC by u_2^*. Hence u_2^* may be selected as a member of C. □

From Lemma 9 and Lemma 10, it is observed that either u_1^* or u_2^* may be selected as a member of C. But our aim is to find C with minimum cardinality. So, under the condition of Lemma 10, u_2^* is to be selected instead of u_1^*.

If u_1^* be selected as a member of C at any stage then in the next stage either u_{l+2}^* or u_{l+3}^* is to be selected as a member of C. The selection depends on same results which are considered in the next section.

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

parent	if $u, v \in V$, in TIT, $level(u) = j$, $level(v) = j+1$ and $(u, v) \in E$ then $parent(v) = u$,
gparent	if $parent(parent(u)) = v$ then $gparent(u) = v$,
l	the level number at any stage,
u_l^*	the vertex on the main path at level l,
X_l	the set of vertices at level l of TIT which are greater than u_l^*, i.e., $X_l = \{v : v > u_l^*$ and $v \in N_l\}$,
Y_l	the set of vertices at level l of TIT which are less than u_l^*, i.e., $Y_l = \{v : v < u_l^*$ and $v \in N_l\}$,
w_l	the least vertex of the set Y_l, *i.e.*, $w_l = min\{v : v \in Y_l\}$.

It may be noted that $X_l \cap Y_l = \phi$ and $N_l = X_l \cup Y_l \cup \{u_l^*\}$.

3.2 Relation between the vertices of N_l and N_{l+1}

Lemma 11 *If $v \in \cup_{i=0}^{1} X_{l+i}$ then $d(v, u_l^*) \le 2$.*

Proof. To prove this lemma, we refer the TIT of Figure 6. From definition of X_l it follows that $u_l^* < v$ for all $v \in X_l$ and for all l.

Let v be any vertex of X_{l+1}, *i.e.*, $v \in X_{l+1}$. Then $u_{l+1}^* < v < u_l^*$. Since $(u_{l+1}^*, u_l^*) \in E'$, therefore, either $(u_{l+1}^*, v) \in E'$ or $(u_l^*, v) \in E'$ (by Lemma 1). If $(u_{l+1}^*, v) \in E'$ then $d(u_l^*, v) = 2$ (as $u_l^* \to u_{l+1}^* \to v$) or if $(u_l^*, v) \in E'$ then $d(u_l^*, v) = 1$ and hence $d(u_l^*, v) \leq 2$.

Again, let $v' \in X_l$. Then $u_l^* < v' < u_{l-1}^*$. Since $(u_l^*, u_{l-1}^*) \in E'$, therefore, either $(u_l^*, v') \in E'$ or $(u_{l-1}^*, v') \in E'$ (by Lemma 1). Similarly, $d(u_l^*, v') \leq 2$.

Thus $d(u_l^*, v) \leq 2$ for all $v \in \cup_{i=0}^{1} X_{l+i}$. □

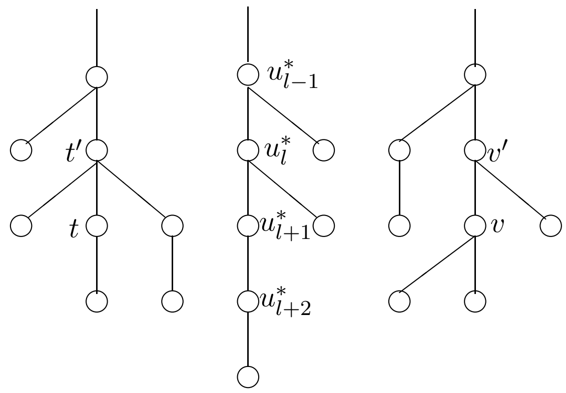

Figure 6: A part of a TIT.

Lemma 12 *If* $t \in \cup_{i=0}^{1} Y_{l+i}$ *then either* $d(t, u_l^*) \leq 2$ *or* $d(t, u_{l+2}^*) \leq 2$.

Proof. To prove this lemma, we refer Figure 6. Let t be any vertex of Y_{l+1}, *i.e.*, $t \in Y_{l+1}$. If $parent(t) = u_l^*$ then $d(u_l^*, t) = 1$. If $parent(t) \neq u_l^*$ and $(parent(t), u_l^*) \in E'$ then $d(u_l^*, t) = 2$ (as $u_l^* \to parent(t) \to t$). But if $(parent(t), u_l^*) \notin E'$ then it is not necessary that $d(u_l^*, t) \leq 2$. Now, $u_{l+2}^* < t < u_{l+1}^*$. Since, $(u_{l+2}^*, u_{l+1}^*) \in E'$ then by Lemma 1, either $(u_{l+2}^*, t) \in E'$ or $(u_{l+1}^*, t) \in E'$. If $(u_{l+2}^*, t) \in E'$ then $d(u_{l+2}^*, t) = 1$ or if $(u_{l+1}^*, t) \in E'$ then $d(u_{l+2}^*, t) = 2$ (as $t \to u_{l+1}^* \to u_{l+2}^*$) and also $d(t, u_l^*) = 2$ (as $t \to u_{l+1}^* \to u_l^*$). Hence for all $t \in Y_{l+1}$, either $d(t, u_l^*) \leq 2$ or $d(t, u_{l+2}^*) \leq 2$.

Again let $t' \in Y_l$. Now, $u_{l+1}^* < t' < u_l^*$. Since, $(u_{l+1}^*, u_l^*) \in E'$, by Lemma 1 either $(u_{l+1}^*, t') \in E'$ or $(t', u_l^*) \in E'$. If $(u_{l+1}^*, t') \in E'$ then $d(u_l^*, t') = 2$ (as $t' \to u_{l+1}^* \to u_l^*$) or if $(t', u_l^*) \in E'$ then $d(u_l^*, t') = 1$. Hence for all $t' \in Y_l$, $d(u_l^*, t') \leq 2$.

Thus for all $t \in \cup_{i=0}^{1} Y_{l+i}$ then either $d(t, u_l^*) \leq 2$ or $d(t, u_{l+2}^*) \leq 2$. □

From Lemma 11, $d(v, u_l^*) \leq 2$ for all $v \in X_{l+1}$. Now if $v \in X_{l+2}$ and $v' \in X_{l+1}$ then $v < u_{l+1}^* < v'$. By Lemma 1 if $(v, v') \in E'$ then either $(v, u_{l+1}^*) \in E'$ or $(u_{l+1}^*, v') \in E'$. If

$(v, u^*_{l+1}) \in E'$ then $d(v, u^*_l) = 2$ (as $v \to u^*_{l+1} \to u^*_l$) or if $(u^*_{l+1}, v') \in E'$ then $d(v', u^*_l) = 2$ but $d(v, u^*_l) = 3$ (as $u^*_l \to u^*_{l+1} \to v' \to v$).

Combining the results of lemmas 11 and 12, we conclude the following result.

Lemma 13 *All edges $(x, y) \in E'$ where $x, y \in \cup_{i=0}^{2} N_{l+i}$ are 2-NC by either u^*_l or u^*_{l+2} or both.*

From above lemma, if u^*_l is selected as a member of C at any stage then in the next stage one can select u^*_{l+2} or u^*_{l+3} as a member of C.

From lemmas 11 and 12, we conclude another result, which is stated below.

Corollary 1 *If $parent(w_{l+1}) = u^*_l$ then the edge (x, y) where $x, y \in \cup_{i=0}^{1} N_{l+i}$ is 2-NC by u^*_l.*

3.3 Selection of next member of C

Let u^*_l be selected as a member of C in the first stage then either u^*_{l+2} or u^*_{l+3} will be selected as a member of C in the next stage. Now u^*_{l+2} may be selected in the next stage. But our aim is to find the set C with minimum cardinality, therefore we will select u^*_{l+3} if possible. The possible cases are described in the following lemmas.

Lemma 14 *If $parent(w_{l+1}) \neq u^*_l$ then u^*_{l+3} can not be a member of C.*

Proof. If $parent(w_{l+1}) \neq u^*_l$ then the TIT has a branch on the left on the main path. To prove this lemma we consider Figure 7. It may be noted that existence of w_{l+1} implies $Y_{l+1} \neq \phi$.

In this case, $parent(w_{l+1}) < u^*_l$. Now if $(parent(w_{l+1}), u^*_l) \in E'$ then $d(w_{l+1}, u^*_l) = 2$ but if $(parent(w_{l+1}), u^*_l) \notin E'$ then by Lemma 1 $(gparent(w_{l+1}), u^*_l) \in E'$. Therefore, $d(w_{l+1}, u^*_l) = 3$ (as $w_{l+1} \to parent(w_{l+1}) \to gparent(w_{l+1}) \to u^*_l$). Thus the edge $(w_{l+1}, parent(w_{l+1}))$ is not 2-NC by u^*_l. Since, $d(w_{l+1}, u^*_{l+2}) \leq 2$ as $u^*_{l+2} < w_{l+1} < u^*_{l+1}$. Therefore, $d(u^*_{l+3}, parent(w_{l+1})) \geq 3$. Again, the edge $(w_{l+1}, parent(w_{l+1}))$ is not 2-NC by u^*_{l+3}. Hence u^*_{l+3} can not be a member of C. □

But, if $parent(w_{l+1}) = u^*_l$ then some times one can select the vertex u^*_{l+3} as a member of C. This selection depends on the nature of the TIT of the trapezoid graph G'.

Lemma 15 *If $parent(w_{l+1}) = u^*_l$ and $X_{l+2} = \phi$ then u^*_{l+3} be a possible member of C.*

Proof. To prove this lemma, we refer Figure 8. The relation $parent(w_{l+1}) = u^*_l$ implies that $d(u^*_l, v) \leq 2$ for all $v \in \cup_{i=0}^{1} N_{l+i}$ (by Corollary 1). So the edge (x, y), $x \in N_{l+1} \cup N_l$ and $y \in N_{l+1} \cup N_l$ is 2-NC by u^*_l.

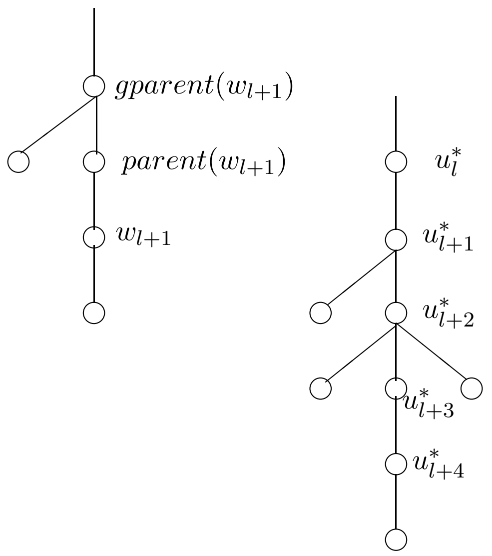

Figure 7: Illustration of lemma 14.

As $X_{l+2} = \phi$, $v \leq u^*_{l+2}$, for all $v \in N_{l+2}$, *i.e.*, $u^*_{l+3} < v < u^*_{l+2}$, for all $v \in N_{l+2}$. Again $(u^*_{l+3}, u^*_{l+2}) \in E'$, so by Lemma 1 either $(v, u^*_{l+2}) \in E'$ or $(v, u^*_{l+3}) \in E'$. If $(v, u^*_{l+2}) \in E'$ then $d(v, u^*_{l+3}) = 2$ (as $v \to u^*_{l+2} \to u^*_{l+3}$) or if $(v, u^*_{l+3}) \in E'$ then $d(v, u^*_{l+3}) = 1$. Thus $d(v, u^*_{l+3}) \leq 2$ for all $v \in N_{l+2}$. Also $d(v, u^*_{l+3}) \leq 2$ for all $v \in N_{l+3}$. So the edge (x, y), $x \in N_{l+2} \cup N_{l+3}$ and $y \in N_{l+2} \cup N_{l+3}$ is 2-NC by u^*_{l+3}. Hence the vertex u^*_{l+3} may be selected as a member of C. □

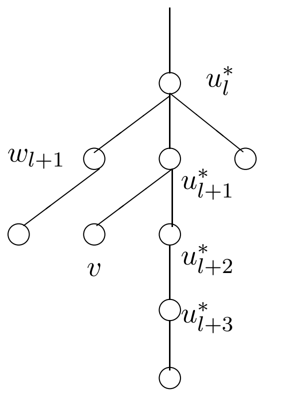

Figure 8: A part of a TIT.

Form the above lemma it follows that if $X_{l+2} = \phi$ then one can select u^*_{l+3} as a possible member of C. But if $X_{l+2} \neq \phi$ then some times one can select u^*_{l+3} as a member of C. The conditions for selecting u^*_{l+3} as a next possible member of C are described below.

Lemma 16 *If* $parent(w_{l+1}) = u_l^*$ *and if* $(u_{l+2}^*, v) \notin E'$ *for at least one* $v \in X_{l+2}$ *where* $X_{l+2} \neq \phi$ *then* u_{l+3}^* *can not be a member of* C*.*

Proof. To prove this lemma, we refer Figure 9. The relation $parent(w_{l+1}) = u_l^*$ implies that $d(u_l^*, v) \leq 2$ for all $v \in \cup_{i=0}^{1} N_{l+i}$ (by Corollary 1). So the edge (x, y), $x \in N_{l+1} \cup N_l$ and $y \in N_{l+1} \cup N_l$ are 2-NC by u_l^*. But the edge (x, y), $x \in N_{l+1}$ and $y \in N_{l+2}$ are not 2-NC by u_l^* as $d(u_l^*, y) \nleq 2$. Now, if $(u_{l+2}^*, v) \notin E'$ for at least one $v \in X_{l+2}$ then the shortest path from u_{l+3}^* to v is $u_{l+3}^* \rightarrow u_{l+2}^* \rightarrow parent(v) \rightarrow v$ (by Lemma 1) and since $v > u_{l+2}^*$, $v \in X_{l+2}$ so it is not necessary that $(v, u_{l+3}^*) \in E'$. Hence $d(v, u_{l+3}^*) = 3$. Thus the edge $(v, parent(v))$, $v \in X_{l+2}$ is not 2-NC by u_{l+3}^*. Therefore, u_{l+3}^* can not be a member of C. □

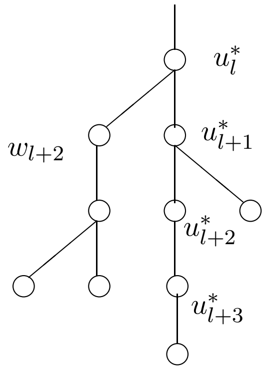

Figure 9: Illustration of lemma 16.

Lemma 17 *If* $parent(w_{l+1}) = u_l^*$ *and* $(u_{l+2}^*, v) \in E'$ *for all* $v \in X_{l+2}$ *but* $parent(v) \neq parent(u_{l+2}^*)$ *for at least one* $v \in X_{l+2}$ *then* u_{l+3}^* *can not be a member of* C*.*

Proof. To prove this lemma, we refer Figure 10. Let $v \in X_{l+2}$ such that $parent(v) \neq parent(u_{l+2}^*)$. In this case, the edge $(v, parent(v))$ is not 2-NC by u_l^* (because, $v < u_{l+1}^* < parent(v)$, so if $(u_{l+1}^*, parent(v)) \in E'$ then $d(v, u_l^*) = 3$). Now, if $(v, u_{l+2}^*) \in E'$ then $d(v, u_{l+3}^*) = 2$ but $d(u_{l+3}^*, parent(v)) = 3$. So the shortest path from u_{l+3}^* to $parent(v)$ is $u_{l+3}^* \rightarrow u_{l+2}^* \rightarrow v \rightarrow parent(v)$. Therefore, the edge $(v, parent(v))$ is not 2-NC by u_{l+3}^*. Hence u_{l+3}^* can not be member of C. □

Lemma 18 *If* $parent(w_{l+1}) = u_l^*$ *and* $(u_{l+2}^*, u) \in E'$ *for all* $u \in X_{l+2} \cup Y_{l+1}$*,* $(v, t) \in E'$ *for at least one* $v \in X_{l+2}$ *and* $t \in Y_{l+1}$ *and* $parent(v) = parent(u_{l+2}^*)$ *for all* $v \in X_{l+2}$ *then* u_{l+3}^* *is a possible member of* C*.*

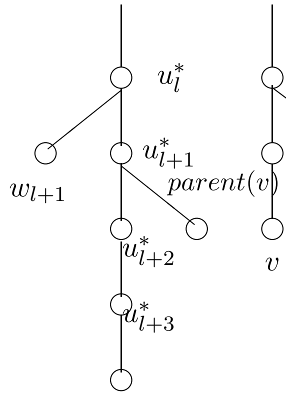

Figure 10: A part of a TIT.

Proof. To prove this lemma, we refer Figure 11. Since $(u_{l+2}^*, u) \in E'$ for all $u \in X_{l+2} \cup Y_{l+1}$ then the edge (x, y), $x \in N_{l+1} \cup N_{l+2}$ and $y \in N_{l+1} \cup N_{l+2}$ is 2-NC by u_{l+3}^* (as $u_{l+3}^* \to u_{l+2}^* \to x$ and $u_{l+3}^* \to u_{l+2}^* \to y$). Also the edge $(parent(u_{l+2}^*), v)$, $v \in X_{l+2}$ is 2-NC by u_{l+3}^* (as $d(parent(u_{l+2}^*), u_{l+3}^*) = 2$, $d(v, u_{l+2}^*) = 2$). Again the edge (t, t'), $t \in Y_{l+1}$, $t' \in Y_{l+2}$ is 2-NC by u_{l+3}^* (as $d(u_{l+3}^*, t) = 2$ and $d(u_{l+3}^*, t') \leq 2$). Hence u_{l+3}^* is a possible member of C. $\square$

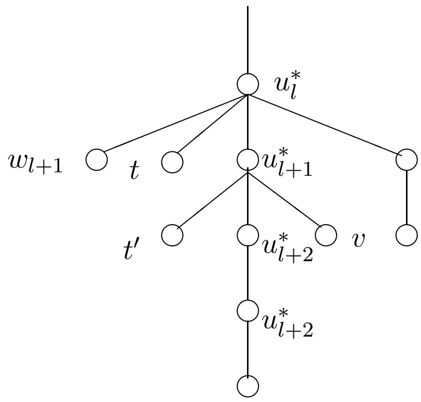

Figure 11: Illustration of lemma 18.

Lemma 19 *If $parent(w_{l+1}) = u_l^*$ and $(u_{l+2}^*, v) \in E'$ and $parent(v) = parent(u_{l+2}^*)$ for all $v \in X_{l+2}$, $(v, t) \in E'$, for all $v \in X_{l+2}$, $t \in Y_{l+1}$ and $(u_{l+2}^*, t) \notin E'$ for at least one $t \in Y_{l+1}$ then u_{l+3}^* can not be a member of C.*

Proof. To prove this lemma, we refer Figure 12. Since $(u_{l+2}^*, v) \in E'$, $v \in X_{l+2}$ then the shortest path from u_{l+3}^* to v is $u_{l+3}^* \to u_{l+2}^* \to v$ and $d(u_{l+3}^*, v) = 2$. But, the shortest path from u_{l+3}^* to t is $u_{l+3}^* \to u_{l+2}^* \to parent(u_{l+2}^*) \to t$ (since $(u_{l+2}^*, t) \notin E'$, then by Lemma 1,

$(parent(u^*_{l+2}), t) \in E'$). So, $d(u^*_{l+3}, t) = 3$. Therefore, the edge (v, t), $v \in X_{l+2}$, $t \in Y_{l+1}$ is not 2-NC by u^*_{l+3}. Hence u^*_{l+3} can not be a member of C. □

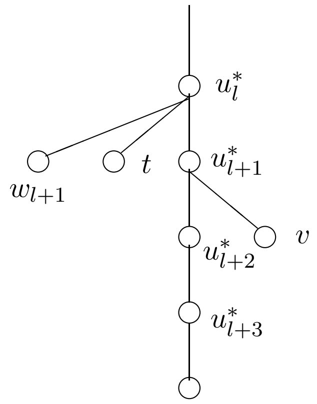

Figure 12: A part of a TIT.

Lemma 20 *If $parent(w_{l+1}) = u^*_l$ for all $v \in X_{l+2}$, $(u^*_{l+2}, v) \in E'$ and $parent(v) = parent(u^*_{l+2})$ and $(v, t) \notin E'$, for all $v \in X_{l+2}$, $t \in Y_{l+1}$ then u^*_{l+3} can not be a member of C.*

Proof. To prove this lemma, we refer Figure 13. Since $(u^*_{l+2}, v) \in E'$, for all $v \in X_{l+2}$ then the edge (v_1, v_2), $v_1, v_1 \in X_{l+2}$ is 2-NC by u^*_{l+3} (as $d(v, u^*_{l+3}) \leq 2$). Let $u \in Y_{l+2}$. Since $u^*_{l+3} < u < u^*_{l+2}$ and $(u^*_{l+3}, u^*_{l+2}) \in E'$ then by Lemma 1, either $(u, u^*_{l+2}) \in E'$ or $(u, u^*_{l+3}) \in E'$. Therefore, $d(u, u^*_{l+3}) \leq 2$ for all $u \in Y_{l+2}$. Again $u^*_{l+2} < t < u^*_{l+1}$, $t \in Y_{l+1}$ and $(u^*_{l+2}, u^*_{l+1}) \in E'$ then either $(t, u^*_{l+2}) \in E'$ or $(t, u^*_{l+1}) \in E'$. If $(t, u^*_{l+2}) \in E'$ then $d(t, u^*_{l+3}) = 2$ but if $(t, u^*_{l+1}) \in E'$ then $d(t, u^*_{l+3}) = 3$. Therefore the edge (u, t) is not 2-NC by u^*_{l+3}. Hence u^*_{l+3} can not be a member of C. □

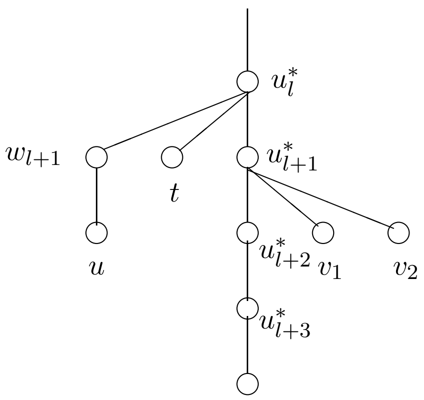

Figure 13: Illustration of lemma 20.

Lemma 21 *If $X_{l+2} = \phi$ and $Y_{l+1} = \phi$ then u^*_{l+3} is a possible member of C.*

Proof. To prove this lemma, we refer Figure 14. Let $t \in Y_{l+2}$ and $t' \in Y_{l+1}$. Since $u^*_{l+3} < t < u^*_{l+2}$ and $(u^*_{l+2}, u^*_{l+3}) \in E'$ then by Lemma 1, either $(u^*_{l+3}, t) \in E'$ or $(t, u^*_{l+2}) \in E'$. Hence $d(t, u^*_{l+3}) \leq 2$ and also $d(t', u^*_{l+3}) \leq 2$. Thus the edge (t, t') is 2-NC by u^*_{l+3}. Hence u^*_{l+3} is a possible member of C. $\square$

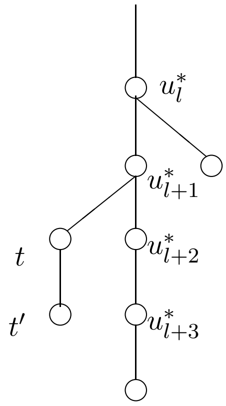

Figure 14: Illustration of lemma 21.

Lemma 22 *If $Y_{l+1} = \phi$ and $(u^*_{l+2}, v) \notin E'$ for at least one $v \in X_{l+2}$ then u^*_{l+2} can not be a member of C.*

Proof. To prove this lemma, we refer Figure 15. If $(u^*_{l+2}, v) \notin E'$ for at least one $v \in X_{l+2}$ then the shortest path from u^*_{l+3} to v is $u^*_{l+3} \rightarrow u^*_{l+2} \rightarrow parent(u^*_{l+2}) \rightarrow v$. Therefore,$d(u^*_{l+3}, v) = 3$. Hence the edge (u, v), $u \in X_{l+1}$ and $v \in X_{l+2}$ is not 2-NC by u^*_{l+3}. Thus u^*_{l+3} can not be a member of C. $\square$

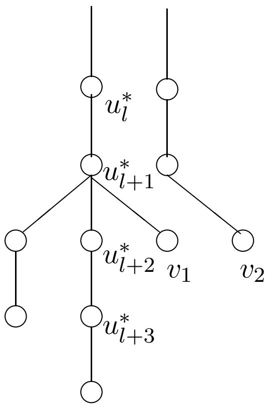

Figure 15: A part of a TIT.

Lemma 23 *If $Y_{l+1} = \phi$, $(u^*_{l+2}, v) \in E'$ for all $v \in X_{l+2}$ and $parent(v) \neq parent(u^*_{l+2})$ for at least one $v \in X_{l+2}$ then u^*_{l+3} can not be a member of C.*

Proof. To prove this lemma, we refer Figure 15. Without loss of generality, we assume that $(u^*_{l+2}, v_2) \in E'$ and $parent(v_2) \neq parent(u^*_{l+2})$, $v_2 \in X_{l+2}$. Since $parent(v_2) \neq Parent(u^*_{l+2})$, $(u^*_{l+2}, parent(v_2)) \notin E'$ as $parent(u^*_{l+2}) < parent(v_2)$. Now $v_2 < parent(u^*_{l+2}) < parent(v_2)$ and $(v_2, parent(v_2)) \in E'$ then either $(v_2, parent(u^*_{l+2})) \in E'$ or $(parent(u^*_{l+2}), parent(v_2)) \in E'$. Therefore, $d(u^*_{l+3}, parent(v_2)) = 3$ and $d(u^*_{l+3}, v_2) = 2$. Hence the edge $(v_2, parent(v_2))$ is not 2-NC by u^*_{l+3}. Thus u^*_{l+3} can not be a member of C. □

Lemma 24 *If $Y_{l+1} = \phi$, $(u^*_{l+2}, v) \in E'$ for all $v \in X_{l+2}$ and $parent(v) = parent(u^*_{l+2})$ for all $v \in X_{l+2}$ then u^*_{l+3} may be a possible member of C.*

Proof. To prove this lemma, we refer Figure 16. Since $(u^*_{l+2}, v) \in E'$ for all $v \in X_{l+2}$, $d(u^*_{l+3}, v) = 2$ (as $u^*_{l+3} \to u^*_{l+2} \to v$). Also, $d(u^*_{l+3}, t) \leq 2$ for all $t \in Y_{l+2}$. Again, $Y_{l+1} = \phi$ and $parent(v) = parent(u^*_{l+2})$ for all $v \in X_{l+2}$. So the edge $(parent(u^*_{l+2}), u)$, $u \in N_{l+2}$ is 2-NC by u^*_{l+3}.

Again, the edge (v, v'), $v \in X_{l+2}$, $v' \in X_{l+3}$ also 2-NC by u^*_{l+3} (since $d(u^*_{l+3}, v') \leq 2$ and $d(u^*_{l+3}, v) = 2$). Hence u^*_{l+3} may be a possible member of C. □

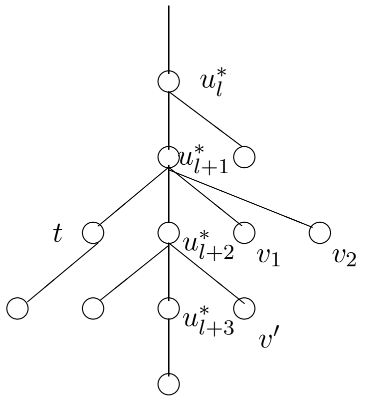

Figure 16: Illustration of lemma 24.

4 Algorithm and its complexity

From the above lemmas it is observed that if u^*_l is selected as a member of C at any stage then either u^*_{l+2} or u^*_{l+3} will be selected as a member of C at next stage. Also, we observed that the vertex u^*_{l+2} may be selected at any stage. But, our aim is to find the set C such that $|C|$ is minimum. To find C with minimum cardinality we will select u^*_{l+3} if possible. All possible cases for selection of the members of C are already presented in terms of lemmas.

4.1 A procedure to compute the next member of C

The procedure NEXTMEMBER is formally presented in the following which computes the level L of the next vertex of u^*_L of C, if the level l of the currently selected vertex u^*_L is supplied.

Procedure NEXTMEMBER(l, L)

//This procedure computes the level L such that u^*_L will be the next member of C where as u^*_l is the currently selected vertex of C. The sets X_i, Y_i and the array u^*_i, $i = 1, 2, \ldots, h$, h is the height of the tree $T(G')$, are known globally.//

Initially $L = l + 2$;
If $Y_{l+1} = \phi$ then
  if $X_{l+2} = \phi$ then $L = l + 3$; (Lemma 21)
  elseif for all $v \in X_{l+2}$, $parent(v) = parent(u^*_{l+2})$ and $(u^*_{l+2}, v) \in E'$ then
    $L = l + 3$; (Lemma 24)
  endif;
else $//Y_{l+1} \neq \phi//$
  if $parent(w_{l+1}) = u^*_l$ then
    if $X_{l+2} = \phi$ then $L = l + 3$; (Lemma 15)
    elseif for all $v \in X_{l+2}$, $parent(v) = parent(u^*_{l+2})$, $(u^*_{l+2}, v) \in E'$ and
      if $(v, t) \in E'$ for some $v \in X_{l+2}$, $t \in Y_{l+1}$ and
      $(u^*_{l+2}, t) \in E'$ then $L = l + 3$; (Lemma 18)
    endif;
  endif;
endif;
return L;
end NEXTMEMBER

Now, in the next section we present the complete algorithm to find a minimum 2-NC set on trapezoid graphs. Using the procedure NEXTMEMBER, we can compute the 2-NC set.

4.2 Algorithm and its time and space complexities to find 2-neighbourhood-covering set

In the following, we design the algorithm 2NC to compute the 2-neighbourhood-covering set of a trapezoid graph.

Algorithm 2NC

Input: A trapezoid graph G and its trapezoid representation.

Output: Minimum cardinality 2-neighbourhood-covering set C.

Step 1: Construct a trapezoid graph $G' = (V', E')$ and its interval representation.

Step 2: Construct a trapezoid interval tree $T(G')$.

Step 3: Compute the vertices on the main path of the tree $T(G')$ and let them u_i^*, $i = 1, 2, \ldots, h$; h is the height of the tree $T(G')$.

Step 4: Compute the sets X_i, Y_i, $i = 1, 2, \ldots, h$.

Step 5: If $(u_1^*, v) \in E'$ for all $v \in X_1 \cup Y_1$ then

    $l = 1$ else $l = 2$;

  endif;

  $C = C \cup \{u_l^*\}$

Step 6: Repeat

  Call NEXTMEMBER (l, L); //Find level L for the next vertex of C//

    $l = L$;

    $C = C \cup \{u_l^*\}$;

 Until $(|h - l| \leq 1)$;

end 2NC.

For the graph of Figure 2, 2-neighbourhood-covering set is $C = \{12, 3\}$. Therefore, the graph of Figure 1, the 2-neighbourhood-covering set is also $\{12, 3\}$.

The vertices of $T(G')$ are the vertices of G'. Therefore, the vertices of $T(G')$ are also the vertices of G. The sets N_i, $i = 1, 2, \ldots, h$ are mutually exclusive and the vertices of each N_i are consecutive integers. Again the sets X_i and Y_i, $i = 1, 2, \ldots, h$ are also mutually exclusive, *i.e.*, $X_i \cap X_j = \phi$, $Y_i \cap Y_j = \phi$, for $i \neq j$ and $i, j = 1, 2, \ldots, h$ and $X_i \cap Y_j = \phi$, $i, j = 1, 2, \ldots, h$, Moreover, $N_i = X_i \cup Y_i \cup \{u_i^*\}$, $i = 1, 2, \ldots, h$. The vertices of each X_i and Y_i are also consecutive integers. So, only the lowest and highest numbered vertices are sufficient to maintain the sets X_i, Y_i, N_i, $i = 1, 2, \ldots, h$. Hence we will store only the lowest and highest numbered vertices corresponding the sets X_i, Y_i, N_i instead of all vertices. If any set is empty then the lowest and highest numbered vertices may be taken as 0. It is obvious that $|\cup_{i=1}^{n} N_i| = n$. In the procedure

NEXTMEMBER, only the vertices of the sets N_l, N_{l+1} and N_{l+2} are considered to process them. The total number of vertices of these sets is $|\cup_{i=0}^{2} N_{l+i}|$ and the subgraph induced by the vertices $\cup_{i=0}^{2} N_{l+i}$ is a part of the tree $T(G')$. So the total number of edges in this portion is less then or equal to $|\cup_{i=0}^{2} N_{l+i}| - 1$. Hence one can conclude the following result.

Theorem 1 *The time complexity of the procedure NEXTMEMBER(l, L) is $O(|\cup_{i=0}^{2} N_{l+i}|)$.*

Time complexity to compute the 2-neighbourhood-covering set of a trapezoid graph is computed in the following theorem.

Theorem 2 *The 2-neighbourhood-covering set of a trapezoid graph with n vertices can be computed in $O(n)$ time.*

Proof. The TIT $T(G')$ of a trapezoid graph G' can be computed in $O(n)$ time. Since the main path starting from the vertex 1 and ending at the vertex n, all the vertices u_l^*, $i = 1, 2, \ldots, h$ on the main path can be identified in $O(n)$ time. The level of each vertex of $T(G')$ can be computed in $O(n)$ time. The sets X_i and Y_i, $i = 1, 2, \ldots, h$ can be computed in $O(n)$ time (Step 4). Each iteration of repeat-until loop takes $O(|\cup_{i=0}^{2} N_{l+i}|)$ time for a given l. The algorithm 2NC calls the procedure NEXTMEMBER for $|C|$ time and each time the value of l is increased by 2 or 3. Step 6 of the algorithm 2NC takes $O(\left|\cup_{i=0}^{h} N_i\right|) = O(n)$ time. Hence overall time complexity is $O(n)$. □

The following theorem gives the space complexity of the algorithm 2NC.

Theorem 3 *The space complexity of the algorithm 2NC is $O(n)$.*

Proof. The n trapezoids T_i $(= [a_i, b_i, c_i, d_i])$ and n intervals can be stored using $O(n)$ space. The TIT $T(G')$, the sets X_i, Y_i and the vertices u_i^*, $i = 1, 2, \ldots, h$ can be stored using $O(n)$ space. Also $|C|$ is equal to $O(n)$. Hence the total space complexity is $O(n)$. □

References

[1] D.Bera, M.Pal and T.K.Pal, An efficient algorithm to generate all maximal cliques on trapezoid graphs, *Intern. J. Computer Math.*, 79 (10)(2002) 1057-1065.

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

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

[4] M. C. Golumbic, *Algorithmic Graph Theory and Prefect Graphs*, Academic Press, New York, 1980.

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

[6] M. Pal and G. P. Bhattacharjee, The parallel algorithms for determining edge-packing and efficient edge dminating sets in interval graphs, *Parallel Algorithms and Applications*, 7 (1995) 193-207.

[7] P. K. Ghosh and M. Pal, An efficient algorithm to find the maximum matching on trapezoid graphs, *Journal of the Korea Society for Industrial and Applied Mathematics-IT Series*, 9 (2) (2005) 13-20.

[8] P. K. Ghosh and M. Pal, Optimal sequential algorithm for computing diameter and centre of a trapezoid graph, *Communicated*.

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

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