**AMO - Advanced Modeling and Optimization, Volume 15, Number 1, 2013**

# An Efficient Algorithm to Solve L(0,1)-Labelling Problem on Interval Graphs

***Satyabrata Paul and Madhumangal Pal***
Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore – 721102, INDIA.
e-mail: {satyamath09, mmpalvu}@gmail.com

***Anita Pal***
Department of Mathematics
National Institute of Technology Durgapur, Durgapur-713209, INDIA.
e-mail: anita.buie@gmail.com

**Abstract**

$L(0,1)$-labelling of a graph $G=(V,E)$ is a function $f$ from the vertex set $V(G)$ to the set of non-negative integers such that adjacent vertices get number zero apart, and vertices at distance two get distinct numbers. The goal of $L(0,1)$-labelling problem is to produce a legal labelling that minimize the largest label used. Since $L(0,1)$-labelling problem is NP-complete for general graph, we investigate the problem for interval graph and present an efficient algorithm for finding the $L(0,1)$-labelling of the said graph.

***Keywords:*** Frequency assignment, $L(0,1)$-labelling, interval graph

***2010 Mathematics Subject Classification:*** 05C78, 68R10, 68W05

## 1. Introduction

In radio frequency assignment, the aim is to assign radio frequency to transmitters at different locations without causing interference. The problem is related to graph coloring problem where the vertices of a graph represent the transmitters and the edges indicate possible interferences between adjacent vertices. The frequencies are modeled as non-negative integers, which we shall refer to as labels or colours.

$L(0,1)$-labelling problem is a special type of graph labelling problem. It is the problem of assigning radio frequencies to transmitters such that transmitters that are closed (distance two apart) receive different frequencies and transmitters that are very close together (distance one apart) receive frequencies zero apart. The span of $L(0,1)$-labelling is the difference between highest and lowest labeled used. The aim is to minimize the span. The minimum span over all possible labelling functions is denoted by $\lambda_{0,1}(G)$ and is called $\lambda_{0,1}$ number of $G$.

In the area of graph labelling problem there are two special type of collisions (frequency interference) have been studied, namely direct collisions and hidden collisions. In direct collisions, a radio station and its neighborhood must have different frequencies, so their signals will not collide. This is nothing but a normal vertex coloring or $L(0,1)$-labelling problem. In hidden collisions, a radio station must receive signals of the same frequencies from any of its neighbour. Thus the only requirement here is that for every station, all its neighbour must have distinct frequencies or labels, but there is no requirement on the label of the station itself. Bertossi et al. [2] studied the case of avoiding hidden collision in the multihop radio networks. To avoid the hidden collision from its adjacent stations, we require distinct labels for its intermediate adjacent stations. Here we suppose that there is a little direct collision in the system. Direct collision is so week that we can ignore it. Hence we allow the same labels for two adjacent stations. Therefore, this problem can be formulated as $L(0,1)$-labelling problem.

Frequency assignment problem has been widely studied in the past [1, 4, 5, 7, 8, 18]. We focus our attention on $L(0,1)$-labelling of graphs. Different bounds for $\lambda_{0,1}(G)$ were obtained for various type of graphs. The upper bound of $\lambda_{0,1}(G)$ of any graph $G$ is $\Delta^2-\Delta$ [9], where $\Delta$ is the degree of the graph. In [3], Bodlaender et al. compute upper bounds for graphs of treewidth bounded by $t$ proving that $\lambda_{0,1}(G)\le t\Delta-t$. They also shown that $L(0,1)$-labelling number of a permutation graph not exceed $2\Delta-2$. In [2], the NP-completeness result for the decision version of the $L(0,1)$-labelling problem is derived when the graph is planar by means of a reduction from 3-vertex colouring of straight-line planer graph. Very recently Khan et al. [10] investigated the problem on cactus graph. In this paper, we investigate the problem on interval graph.

This paper is organized as follows. Some preliminaries are discussed in Section 2. Then an algorithm of $L(0,1)$-labelling of interval graph and some results related to the algorithm are presented in Section 3. Finally, in Section 4 conclusions are made.

## 2. Preliminaries and notations

Throughout the paper we consider finite undirected interval graphs without multiple edges or loops. The vertex set (edge set) of a graph $G$ is denoted by $V(G)$ ($E(G)$ respectively).

Let $d(x,y)$ be denot the distance between $x$ and $y$ in a graph $G$, which is the length of the shortest path joining $x$ and $y$.

Let $N(v)=\{u\in V(G):(u,v)\in E(G)\}$ denotes the set of neighbours called the open neighborhood of the vertex $v$. The set $N[v]=N(v)\cup\{v\}$ denoted the closed neighborhood of $v$. The symbol $n$ is reserved for the number of vertices of the graph $G=(V,E)$.

**Definition 1.** *(Interval graph) An undirected graph $G=(V,E)$ is an interval graph if the vertex set $V$ can be put into one-to-one correspondence with a set of intervals $I$ on the real line $R$ such that two vertices are adjacent in $G$ iff their corresponding intervals have non-empty intersection.*

The intervals and the vertices of an interval graph are the same things. The set $I$ is called an interval representation of G and G is referred to as the intersection graph of $I$. Here, we assume that the input graph is given by an interval representation $I$ which is the set of $n$ sorted intervals labelled by $1,2,\dots,n$.

Let $I=\{I_1,I_2,\dots,I_n\}$, where $I_j=[a_j,b_j]$, $j=1,2,\dots n$; be the interval representation of the given interval graph $G=(V,E)$, $V=\{1,2,\cdots,n\}$, $a_j$ and $b_j$ are respectively the left and the right end points of the interval $I_j$. Without any loss of generality, we assume that each interval contains both its end points and that no two intervals share a common end point. Also, we assume that the intervals in $I$ are indexed by increasing right end points, that is, $b_1<b_2<\cdots<b_n$. This indexing is known as IG ordering.

Adjacency of two intervals (also vertices) can be tasted using the result of the following lemma.

**Lemma 1.** *[17] If $[a_i,b_i]$ and $[a_j,b_j]$ are endpoints of the vertices $i$ and $j$ respectively then $i$,*

*$j$ are non-adjacent if and only if $b_i < a_j$ or $b_j < a_i$.*

Interval graphs and many of its applications are discussed extensively in [6, 11, 12, 13, 14, 15, 16, 17]. This graph satisfies lot of intersecting properties. We just point out some of them.

**Lemma 2.** *For the vertices $i, j, k$ of an interval graph $G$, if $k < i < j$ and $d(i, j) > 2$, then $d(k, j) > 2$.*

**Proof.** Let $G$ be an interval graph and $d(i, j) > 2$, where $j > i$. Also we assume that the interval in $I$ are indexed by increasing right end points. So when $k < i$ then $b_k < b_i$ (right end point of the interval $I_k$ is less than the right end point of the interval $I_i$). Thus the distance from $k$ to $j$ is obviously grater than 2 as $d(i, j) > 2$. Hence the result. □

**Lemma 3. (***[14]) The graph $G$ is an interval graph if and only if there exist an ordering of its vertices $v_1 < v_2 < v_3 < .... < v_n$ such that $v_i < v_j < v_l$ and $(v_i, v_l) \in E$ then $(v_j, v_l) \in E$.*

**Lemma 4.** *([6]) The maximal clique of an interval graph $G$ can be linearly ordered such that for every vertex $v \in G$, the maximal clique containing $v$ occurs consecutively.*

We now define some special types of sets of vertices.

**Definition 2.** *Let $B_\alpha$ is the set of 2-nbd vertices of $\alpha$ with index greater than $\alpha$.*
i.e., $B_\alpha = \{j \in V(G) \mid d(j, \alpha) = 2 \text{ and } j > \alpha\}$.

For example, for the graph of Figure 1, $B_5 = \{7,10\}$.

**Definition 3.** *$B_\alpha^i$ is the set of 2-nbd vertices of $\alpha$ passing through $i$ and index greater than $\alpha$.*
i.e., $B_\alpha^i = \{j \in V(G) \mid d(j, \alpha) = 2$ and the path joining $j$ and $\alpha$ passing through $i$ and $j > \alpha\}$.

For the graph of Figure 1, $B_5^8 = \{7,10\}$ but $B_5^6 = \phi$.

**Definition 4.** *For an interval graph $G$ we define a set $S_i$, where $i \in V(G)$ such that*
(i) all the vertices of $S_i$ are adjacent to $i$.
(ii) no two vertices of $S_i$ are adjacent.
(iii) each $S_i$ is maximal.

Some characteristic of the set $S_i$ are state below.
(i) $S_i$, for all $i = 1, 2 \ldots, n$ is not a singleton set.
(ii) The set $S_i$ for some $i \in V(G)$ may not be unique, but the cardinality of the set is unique. For example, in Figure 1, $S_{10} = \{7,9\}$ or $\{7,12\}$ or $\{8,9\}$ or $\{8,12\}$. That is $S_{10}$ is not unique but

$|S_{10}|=2$.

(iii) The vertices of $S_i$ are independent, follows from the condition (ii)

(iv) If $x, y \in S_i$, then $d(x,i)=d(y,i)=1$, but $d(x,y)=2$.

**Lemma 5.** *If $S_i=\phi$, for all $i \in V(G)$, then $G$ is complete.*

**Proof.** If $S_i=\phi$ for some $i \in V(G)$, then there does not exist at least two vertices $x$ and $y$ in $S_i$ such that $d(x,y)=2$ via $i$. That is, the vertices of $G$ are adjacent to each other. This is true for all $i \in V(G)$. Hence the result. □

In the next lemma we established a relation between $\lambda_{0,1}(G)$ and $\max|S_i|$, $i=1,2,\ldots,n$ for an interval graph.

**Lemma 6.** *Let $k=\max_{i\in V(G)}|S_i|$, $i=1,2,\ldots,n$. Then at least $k$ labels are needed to label an interval graph by $L(0,1)$-labelling.*

**Proof.** Let $\alpha \in V(G)$ such that $|S_\alpha|=\max_{i\in V(G)}|S_i|=k$. Clearly $S_\alpha \cup \alpha$ forms a subgraph of $G$. Thus when we label this subgraph by $L(0,1)$-labelling then any one member of $S_\alpha$ and $\alpha$ takes same label and all other member(s) get distinct labels. Thus exactly $k$-labels (namely $0,1,\ldots,k-1$) needed to label the subgraph $S_\alpha \cup \alpha$. Hence $\lambda_{0,1}(G) \geq k-1$.

From this lemma one can conclude that the lower bound of $\lambda_{0,1}(G)$ of interval graph is $\max_{i\in V(G)}|S_i|-1$. □

The algorithm max $S$, generates all $|S_i|, i=1,2,\ldots,n$, and then computes the value of max $\{|S_i|: i=1,2,\ldots,n\}$.

**Algorithm max $S$**

**Input:** Interval represent of the interval graph.

**Output:** Maximum $|S_i|$, for $i=1,2,\ldots,n.$

**Step** 1. Repeat steps 1 to 3 for $i=1,2,\ldots,n.$

Choose $[a_i,b_i]$ // $a_i$ and $b_i$ represent left and right end points of the Interval $i$ respectively//

**Step** 2. Search an interval $[b_x,a_y]$ on the real line $R$ such $[b_x,a_y]\subset[a_i,b_i]$

**Step** 3. If there exist no $x$ and $y$ such that $[b_x,a_y]\subset[a_i,b_i]$, then

set $|S_i|=0$;

else

choose $b_j \in [a_i,b_i]$ such that no other right end points of some intervals appear between $[a_i,b_j]$;

set temp= $j$;

count=1;

**Step** 3.1. Find $B^i_{temp}$ // by definition 3//

**Step** 3.2. If $B^i_{temp} = \phi$ then go to Step 3.3;

else

find min $B^i_{temp}$;

set temp=min $B^i_{temp}$;

count=count+1;

go to Step 3.1;

**Step** 3.3. Set $|S_i|$=count;

**Step** 4. Find maximum of $\{|S_1|,|S_2|,\ldots,|S_n|\}$

**end max** $S$ **.**

### 3. The Algorithm

The strategy of our proposed algorithm is as follows. First of all we partition the vertex set $V(G)$ into a minimum number of disjoint sets (say $p$) in such a way that each set forms a clique. Let $V(G) = \{U_1, U_2, \ldots, U_p\}$ be such a partition, where $U_i \cap U_j = \phi$ for all, $i, j = 1,2,\ldots, p$, $i \neq j$ and each $U_i$ is a clique.

The following algorithm is used to partition the set of vertices.

**Algorithm VP**

**Input:** Interval representation of the interval graph $G$.

**Output:** Partition $\{U_1, U_2, U_3, \ldots U_n\}$ of the vertices of $G$.

**Step 1.** (Initialization)

$S_1 = V$ (Set of vertices of $G$).

$U_1 = N[1]$ // $U_1$ is the first partition.//

Set $i = 1$.

**Step 2.** Compute $S_{i+1} = S_i - U_i$.

If $S_{i+1} = \phi$ then

stop;

else

Compute $minS_{i+1}$;

set $j = minS_{i+1}$;

find $N[j]$;

**Step 3.** Set $U_{i+1} = N[j] \cap S_{i+1}$;

**Step 4.** Set $i = i+1$ and go to step 2;

**end VP**.

Throught the paper, labelling of a partition means the labelling of its vertices by a label and unlabelled partition means the vertices of this partition are not labeled by any label.

After partition of the vertices we label the vertices of the graph using $L(0,1)$-labelling. First we define a set $L=\{0,1,\cdots,k-1\}$, that is, the set of labels which are initially used to label the graph, where $k-1$ is the lower bound of the label (by Lemma 6). Then we label the partitions $U_1,U_2,\cdots,U_k$ by $0,1,\cdots,k-1$ respectively. The remaining partitions (if exist) are labeled one by one in ascending order of their suffixes. The choice of label for an unlabeled partition depend on its previous partition. Let $U_m$ was already been labeled but $U_{m+1}$ is unlabeled, then we choose the label $f(U_m)+1$ (mod $|L|$) for the partition $U_{m+1}$. Then we check the validity of the label $f(U_{m+1})$ for the partition $U_{m+1}$. If the label is not valid(does not fulfill the condition of $L(0,1)$-labelling) for the partition $U_{m+1}$, then we choose another label $f(U_m)+2$ (mod $|L|$) for the partition $U_{m+1}$ and again check the validity of this new label for the partition $U_{m+1}$. If all the labels of the set $L$ are not valid for $U_{m+1}$, then we choose a new label which does not belong to $L$ and assign this new label for the partition $U_{m+1}$.

We now define a P-set as follows.

**Definition 5.** *Suppose $f(U_m)=l$. To check the validity of the label $l$ for the partition $U_m$, we find a partition $U_j$ such that $f(U_j)=l$, where $j<m$ and there exist no partition $U_k$, where $j<k<l$ such that $f(U_k)=l$. We define a set called P-set corresponding to $U_m$ is denoted by $P_j^m$ and is define as*

$$P_j^m=\{u\in U_j : d(u,v)\le 2, \textit{ where } u\in U_j \textit{ and } v\in U_m\}.$$

If no partition $U_j$ for $j<m$ exist then we say that the P-set corresponding to $U_m$ is the null set.

Here we observe two situations stated below.

(I) Situation 1: Let $f(U_i)=l$. To check the validity of the label $l$ for the partition $U_i$, we find $P_j^i$. If $P_j^i\cup U_{j+1}$ does not form a clique and there exist *a vertex* $x\in U_k$, where $k<j<i$ such that $f(x)=f(U_{j+1})$ and $d(x,y)=2$, where $x\in U_k$ and $y\in P_j^i$.

(II) Situation 2: Let $f(U_i)=l$. To check the validity of the label $l$ for the partition $U_i$, we find $P_j^i$. If $P_j^i\cup U_{j+1}$ does not form a clique and there exist *no vertex* $x\in U_k$, where $k<j<i$ such that $f(x)=f(U_{j+1})$ and $d(x,y)=2$, where $x\in U_k$ and $y\in P_j^i$.

When Situation 2 occurs then we define the set $q_m^i$ and $T_i$ as follows.

$$q_m^i=\{w\in V : d(u,v)=2 \text{ passing through } w, \text{ where } u\in P_m^i \text{ and } v\in U_i\}$$

and $Q_m^i$ is the set of all 1-nbd vertices of $q_m^i$. The set $T_x$ is the set of all 2-nbd vertices of $x$.

Assume that $L_x$ is the set of labels of all the vertices of $T_x$.

Finally, we present the formal algorithm to label the vertices of an interval graph using $L(0,1)$-labelling.

**Algorithm L01**

**Input:** Interval represent of the interval graph $G$.
**Output:** Optimal $L(0,1)$ -labeling of $G$.
**Step 1.** Construct a set $L=\{0,1,\cdots,k-1\}$ using the Algorithm max $S$.
**Step 2.** Construct all the partitions using algorithm VP and label the partitions $U_1,U_2,\cdots,U_k$ by $0,1,\cdots,k-1$ respectively. That is, $f(U_i)=i-1$, $i=1,2,\ldots,k$.
**Step 3.** Set $f(U_{k+1})=f(U_k)+1 \pmod{|L|}$.
That is the value of $f(U_{k+1})$ must be equal to one of the values of $f(U_1),\ldots,f(U_k)$.
Suppose $f(U_{k+1})=f(U_s)$, $1\le s\le k$.
**Step 4.** Find the P-set corresponding to $U_{k+1}$. That is, $P_s^{k+1}$.
**Step 5.** If $P_s^{k+1}=\phi$, i.e. if there is no vertex $v_q\in U_s$ such that $d(v_q,v_r)\le 2$ for all $v_r\in U_{k+1}$, then $f(U_{k+1})$ is the valid label for the vertices of $U_{k+1}$, then $f(U_{k+1})$ is valid for $U_{k+1}$.
**Step 6.** If $P_s^{k+1}=U_s$, i.e. if all the vertices of $U_s$ are at a distance 2 from at least one vertex of $U_{k+1}$, then $f(U_{k+1})$ is not valid label for $U_{k+1}$.
**Step 7.** If some vertices of $U_s$ are at a distance 2 from at least one vertex of $U_{k+1}$, then two cases may aries.
**Step 7.1.** If $U_{s+1}\cup P_s^{k+1}$ form a clique then shift the vertices of $P_s^{k+1}$ from $U_s$ to $U_{s+1}$ and set $f(P_s^{k+1})=f(U_{s+1})$.
Now check the validity of the label $f(U_{s+1})$ for the new partition $U_{s+1}$ proceed same as above.
If $f(U_{s+1})$ is valid for $U_{s+1}$ then $f(U_{k+1})$ is valid for $U_{k+1}$.
Otherwise $f(U_{k+1})$ is not valid for $U_{k+1}$.
**Step 7.2.** If $U_{k+1}\cup P_s^{k+1}$ does not form a clique then either Situation 1 or Situation 2 occurs.
**Subcase 7.2.1.** If Situation 1 occurs then $f(U_{k+1})$ is not a valid label for $U_{k+1}$.
**Subcase 7.2.2.** If Situation 2 occurs then
(i) Find $q_s^{k+1}$ and $Q_s^{k+1}$.
(ii) Remove the labels of all the vertices of $Q_s^{k+1}$ and label the vertices of $Q_s^{k+1}$ in the following fashion.
Find the minimum index vertex (say $y$) from the set $Q_s^{k+1}$.
Find $T_y$ and $L_y$.
Set $f(y)=min\{L-L_y\}$.

Now find the next minimum index vertex (unlabelled) of the set $Q_s^{k+1}$ and to label this vertex, follow the same procedure.
If all the vertices of $Q_s^{k+1}$ are labelled by the above way, then $f(U_{k+1})$ is a valid label for $U_{k+1}$.
Otherwise (i.e, if $L - L_y = \phi$ for some $y \in Q_s^{k+1}$) $f(U_{k+1})$ is not a valid label for $U_{k+1}$.

**Step 8.** If the label $f(U_{k+1})$ for $U_{k+1}$ is not valid, then reset $f(U_{k+1})$ as $f(U_{k+1}) + \quad 45pti(mod\,|\,L\,|)$, and repeat the above steps for $i = 2,3,\ldots,k-1$.
If $f(U_{k+1})$ is not valid label then use a new label which does not belongs to $L$. This new label is taken as the least integer which does not belongs to $L$, let it be $j$. Update $L$ as $L \cup \{j\}$ and set $f(U_{k+1}) = j$.

**end L01**

### 3.1. Proof of correctness and time complexity

**Lemma 7.** *If $P_i^j \cup U_{i+1}$ forms a clique, then $f(P_i^j) = f(U_{i+1})$ preserve the $L(0,1)$-labelling condition between the labels of new $U_{i+1}(= U_{i+1} \cup P_i^j)$ and $U_m$, where $m > i+1$.*

**Proof.** For the sake of simplicity, we use the term "right side of a partition $U_x$" means a partition $U_y$, where $y > x$.

Let $f(U_j) = a$ and we now check the validity of the label $l$ for the partition $U_j$. So first we have to find $P_i^j$.

Let $f(U_i) = a$ and $f(U_{i+1}) = b$.

**Case 1:** When there is no partition with label $b$ in the right side of $U_{i+1}$.
In this case shift the vertices of $P_i^j$ from $U_i$ to $U_{i+1}$ and set $f(P_i^j) = f(U_{i+1})$. Thus the label of the new $U_{i+1}(= U_{i+1} \cup P_i^j)$ preserve $L(0,1)$-labelling condition in the right side of new $U_{i+1}$ as there is no partition with label $b(= f(U_{i+1}))$ in the right side of new $U_{i+1}$.

**Case 2:** When there exist at least one partition with label $b$ in the right side of $U_{i+1}$.

Let $U_i, U_{i+1}, \ldots, U_j, U_s$, where $i < i+1 < \ldots < j < s$ be some partition which are already labelled. Again, let $f(U_i) = a, f(U_{i+1}) = b, \ldots, f(U_j) = a, f(U_s) = b$. Here $U_s$ is the first partition labelled by $b$ in the right side of $U_{i+1}$. Now we check the validity of the label $a$ for the partition $U_j$.

Let $P_i^j = \{\alpha_1, \alpha_2, \ldots, \alpha_r\} \subset U_i$.

**Subcase 2.1:** If there exist at least one vertex $x \in U_{i+1}$ such that $x > \alpha_i$, for all $\alpha_i \in P_i^j$. Here two partitions $U_{i+1}$ and $U_s$ are already labelled. So they satisfy the labelling condition. That is, $d(u,v) > 2$, for all $u \in U_{i+1}$ and for all $v \in U_s$. Thus by Lemma 2, $d(\alpha_i, v) > 2$ for all $\alpha_i \in P_i^j$ and $v \in U_s$.

Hence $f(P_i^j)=f(U_{i+1})$ satisfy the labelling condition between the labels of new $U_{i+1}$ and $U_m$, where $m>i+1$.

**Subcase 2.2:** If there exist no vertex $x\in U_{i+1}$ such that $x>\alpha_i$, for all $\alpha_i\in P_i^j$. If possible let $d(\alpha_l,y)\le 2$ for all $\alpha_i\in P_i^j$ and $y\in U_s$. That is, $f(P_i^j)=f(U_{i+1})$ violate the labelling condition between the labels of new $U_{i+1}$ and $U_s$. But $f(U_{i+1})=f(U_s)=b$. That is, they satisfy the labelling condition. Thus this case is possible only when $a_y<a_t$, for all $t\in U_j$ and $y\in U_s$ ($a_y$ and $a_t$ are the left end points of the vertices of $y$ and $t$ respectively). Thus by Algorithm VP initially $y\in U_j$. So when we are going to label the partition $U_j$ then $\alpha_l\in P_i^j$. By Algorithm L01, at that time $\alpha_l$ was transform from $U_i$ to $U_{i+1}$, which contradicts our assumption that $\alpha_l\in U_i$. Thus we conclude that this case (subcase 2.2) cannot occur in our labelling procedure. Hence prove the lemma. □

**Theorem 1.** *Algorithm L01 correctly gives the value of* $\lambda_{0,1}(G)$.

**Proof. Case 1:** If the set $L=\{0,1,\ldots,k-1\}$ is sufficient to label the whole graph, then the theorem is obviously true as by Lemma 6, $k-1$ is the lower bound of $G$ by $L(0,1)$-labelling.

**Case 2:** If we use extra label then we have to show that previous set of labels are not sufficient to label the graph.

Suppose we now going to label the partition $U_t$ by label $l$. We now explain the different cases when $l$ is not valid label for $U_t$.

(a) Suppose $P_i^j=U_i$ occur at any stage when we check the validity of the label $l$ for the partition $U_t$. Now if $f(U_j)$ is valid, then $l$ is valid label for $U_t$.

<u>Subcase 2.1.</u> When all labels are present between $f(U_i)$ and $f(U_j)$, where $1\le i<j$.

In this case there exist a partition $U_x, i<x<j$ such that $f(U_x)=k(\ne l)\in L$. Again $P_i^j=U_i$, i.e, for all $a\in U_i$ and some $b\in U_j, d(a,b)\le 2$ passing through $c\in U_x$. Hence we cannot use the same label for the partition $U_x$ and $U_i$. Thus we cannot assign any $k(\ne l)\in L$ for $U_i$. Thus $f(new\,U_j)$ is not valid for $new\,U_j$ and consequently $f(U_t)\ne l$.

<u>Subcase 2.2.</u> When all the labels are not present between $f(U_i)$ and $f(U_j)$, where $1\le i<j$.

Let $U_i,U_{i+1},\ldots,U_x,U_{x+1},\ldots,U_j$ be some consecutive partitions and let $m+1$-labelled partition is absent between $f(U_i)$ and $f(U_j)$. By subcase 2.1 we cannot update the label of $U_i$ by a label which are present between $f(U_i)$ to $f(U_j)$. So we try to update the label of $U_i$ by $m+1$.

Let $f(U_b)=m+1$, where $b<i$ and $f(U_x)=m, f(U_{x+1})\ne m+1$ and there exist no partition labelled by $m+1$ between $U_b$ and $U_{x+1}$. Now $i<x+1$, thus by Algorithm VP there exist some vertex $p\in U_i$ and $q\in U_{x+1}$ such that $a_p<a_q$, ($a_p$ and $a_q$ are the left end points of $I_p$ and $I_q$ ). Since $m+1$ is not valid for the partition $U_{x+1}$ (as $U_{x+1}\ne m+1$) due to $f(U_b)$. Therefore, $m+1$ is also not valid label for the partition $U_i$ as $a_p<a_q$, where $p\in U_i$ and $q\in U_{x+1}$. So $f(U_i)$

cannot be updated by the label $m+1$. Therefore, we cannot assign $l$ for the partition $U_t$.

(b) When situation 1 occurs then we cannot set $f(p_i^j) = f(U_{i+1})$ as $d(p,q) = 2$ for some $p \in P_i^j$ and some $q \in U_{i+1}$.

Again, from situation 1 there exist some partition $U_k$ such that $f(U_k) = f(U_{i+1})$, where $k < i < j$ and there exist no partition with label $f(U_{i+1})$ between $U_k$ and $U_{i+1}$. So the partitions $U_k$ and $U_{i+1}$ satisfy labelling condition. So all the labels of $L$ are either present or some label are absent between $f(U_k)$ and $f(U_{i+1})$.

If all the labels are present, then the explanation is same as subcase 2.1 of (a) and if some labels are absent then follow the explanation of subcase 2.2 of (a).

(c) If situation 2 occur then we follow step 7 of our algorithm. If all the vertices of $Q_i^j$ are labelled then there is no problem. But if $L - L_y = \phi$ occur for some $y \in Q_i^j$ then we cannot label $y$ by a label which belongs to $L$. But initially all the vertices of $Q_i^j$ was labelled. Now we cannot label $y$ due to $f(new\,U_j)$. So $f(new\,U_j)$ is not a valid label for $new\,U_j$ and consequently $f(U_t) \neq l$.

Here we discuss about a particular label $l$ which is not valid for the partition $U_t$. In this way, when all the labels of the set $L$ are not valid for the partition $U_t$ then we conclude that previous set of labels are not sufficient to label the vertices from $U_1$ to $U_t$. So we introduce a new label for the partition $U_t$.

Therefore, any interval graph label by algorithm L01 gives the correct value of $\lambda_{0,1}$. □

**Theorem 2.** *The algorithm L01 takes $O(n^2 |L|)$ time to label all the vertices of an interval graph, where $n$ is the number of vertices of the graph and $|L|$ is the cardinality of the set of labels $L$.*

**Proof.** In Step 1 of algorithm L01, the set L can be computed from the algorithm $\max S$. In algorithm $\max S$, to find $|S_\alpha|$ for an arbitrary interval $\alpha$, it takes at most $O(n)$ times. Thus the total time complexity of the algorithm $\max S$ is $O(n^2)$.

Adjacency of any two vertices can easily be determined from the interval represent of an interval graph. Unit time required to determine the adjacency of any two vertices. To partition the vertices each $U_i$ takes $|U_i|$ times. Again $\sum_{i=1}^{p} |U_i| = n$, where $p$ is the number of partitions. Therefore the algorithm VP takes $O(n)$ times.

In algorithm L01, label a vertex $x \in V(G)$ for a particular label(say $l$), we have to scan at most all the labeled vertices. Thus to label a vertex for a particular label it takes at most $O(n)$ time. But this label $l$ may not be a valid label for the vertex $x$. So we have to choose another label from the set $L$ and check the validity of this label for $x$. So it takes again $O(n)$ time. In worst case, we have to check the validity of all the labels of the set $L$ for the vertex $x \in V(G)$. Thus algorithm L01 takes at most $O(n|L|)$ time to label a vertex. Therefore, to label all the vertices of $G$, algorithm L01 takes $O(n^2 |L|)$ time, where $|L|$ is the cardinality of the label set $L$. □

### 3.2 Verification of the algorithm

Suppose we are to find the value of $\lambda_{0,1}$ and label the interval graph $G=(V,E)$, where $V=\{1,2,\ldots,14\}$ shown in Figure 1. First of all we have to find the value of $k-1$, i.e, the lower bound of the graph $G$. To find the lower bound, we use algorithm max$|S_i|$.

First set $i=1\in V(G)$ and choose $[a_1,b_1]$. Now we have to find an interval $[b_x,a_y]$ for some $x,y\in V(G)$ such that $[b_x,a_y]\subset[a_1,b_1]$. In fact no $x,y\in V(G)$ can found for $i=1$. So by Step 3, $|S_1|=0$.

For $i=2$, choose $[a_2,b_2]$. In this case there exist an interval $[b_1,a_4]$ such that $[b_1,a_4]\subset[a_2,b_2]$. By step 3, set temp=1 and count=1. By step 3.2, $B_1^2=4$. That is, $B_1^2\neq\phi$ (by Step 3.2). So we find new value of temp. Now temp=min $B_1^2=4$. So count is increased by 1. That is, count=1+1=2. Again by step 3.1, $B_{temp}^2=B_4^2=\phi$. Thus by step 3.3, set $|S_2|=2$.

Similarly $|S_3|=2,|S_4|=2,|S_5|=2,\ldots,|S_{13}|=2,|S_{14}|=0$.

Therefore, by Lemma 2, $k=2$. So 1 is the lower bound of $G$.

Now we follow Algorithm VP to partition the vertex set $V(G)$.

By Step 1, $S_1=\{1,2,3,\ldots,14\}$ and $U_1=N[1]=\{1,2,3\}$ is the first partition.

For $i=1$, $S_2=S_1-U_1=\{4,5,\ldots,13,14\}$. Now $S_2\neq\phi$. So we compute min $S_2$. Here min $S_2=4$. Set $j=4$ and $N[j]=N[4]=\{2,3,4,5,6\}$. By Step 3, $U_2=N[4]\cap S_2=\{4,5,6\}$.

Similarly, $U_3=\{7,8,10\},U_4=\{9,12\},U_5=\{11,13\},U_6=\{14\}$.

Now to label the graph, we follow Algorithm L01.

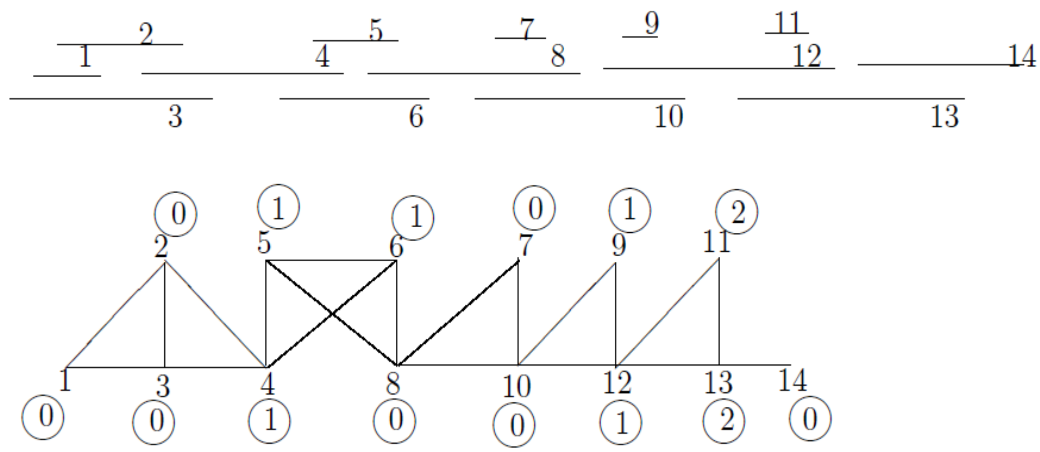


**Figure 1:** interval representation and L(0,1)-labelling of the corresponding interval graph G.

By Step 1, $L=\{0,1\}$. Now we collect all the partitions and set $f(U_1)=0, f(U_2)=1$.

For $U_3$, set $f(U_3)=f(U_2)+1(mod\,|L|)=1+1(mod\ 2)=0$. Now we check whether 0 is a valid label for $U_3$.

First we find $P_1^3$. Here $P_1^3=\phi$. So by Step 4, $f(U_3)=0$ is valid.

For $U_4$ set $f(U_4) = f(U_3) + 1(mod\,|L|) = 0 + 1(mod\ 2) = 1$.

Now we find the P-set corresponding to $U_4$, i.e, $P_2^4$. Here $P_2^4 = \phi$. So by Step 4, $f(U_4) = f(\{9,12\}) = 1$ is valid.

For $U_5$ set $f(U_5) = f(\{11,13\}) = f(U_4) + 1(mod\,(|L|) = 1 + 1(mod\ 2) = 0$

Now to check the validity of the label 0 for the partition $U_5$ we find $P_3^5 = \{10\} \neq \phi$. Now by case 1 of step 7, $\{10\} \cup U_4 = \{9,10,12\}$ forms a clique. So we move the vertex $\{10\}$ from $U_3$ to $U_4$. So new $U_3 = \{7,8\}$ and new $U_4 = \{9,10,12\}$ and set $f(\{10\}) = 1$.

Therefore $f(new U_4) = f(\{9,10,12\} = 1$. Now we check the validity of the label 1 for the partition new $U_4$. Here $P_2^4 = \{5,6\} \neq \phi$. Now $P_2^4 \cup U_3 = \{5,6,7,8,9\}$ does not form a clique. Therefore either situation 1 or situation 2 occur.

Here $\{2,3\} \in U_1$ and $f(\{2,3\}) = f(U_3) = f(\{7,8\})$ and $d(2,5) = d(2,6) = d(3,5) = d(3,6) = 2$. Thus situation 1 occur.

So $f(new U_4) = f(\{9,10,12\}) = 1$ is not valid for the partition new $U_4$ and consequently $f(U_5) = f(\{11,13\}) = 0$ is also not valid. Thus we back to our previous valid labelled partition, i.e, $f(U_3) = f(\{7,8,10\}) = 0$ and $f(U_4) = f(\{9,12\}) = 1$.

By Step 7, set $f(U_5) = f(U_4) + 2(mod\,|L|) = 1 + 2(mod\ 2) = 1$.

Now we check the validity of the label 1 for the partition $U_5$.

Here $P_4^5 = \{9,12\} = U_4$. By Step 6, $f(U_5) = 1$ is not a valid label.

Thus 0 and 1 both are not valid for the partition $U_5$. So by Step 7, we introduce a new label namely 2. Now the new set $L$ is $\{0,1,2\}$ and set $f(U_5) = 2$ by Step 8.

For $U_6$ set $f(U_6) = f(U_5) + 1(mod\,|L|) = 2 + 1(mod\ 3) = 0$.

Now $P_3^6 = \phi$. Thus 0 is valid for the partition $U_6$. Hence $f(U_6) = f(\{14\}) = 0$.

Thus, finally the label of all the vertices of the graph of Figure 1 are

$f(1) = 0, f(2) = 0, f(3) = 0, f(4) = 1, f(5) = 1, f(6) = 1, f(7) = 0, f(8) = 0, f(9) = 1,$
$f(10) = 0, f(11) = 2, f(12) = 1, f(13) = 2$ and $f(14) = 0$

## 4. Conclusion

In this paper, we consider the frequency assignment problem for a particular class of graphs, viz. interval graphs. Computation of $\lambda_{0,1}(G)$ of general graph is NP-hard and also for some other graphs. But, we have shown that the value of $\lambda_{0,1}$ can be computed for interval graph using polynomial time. An $O(n^2\,|L|)$ time algorithm is design to solve this problem on interval graph, where $|L|$ is the cardinality of the st of labels used. We expected that this is not the optimal time and hence there is a scope to reduce the time complexity. We are trying to solve this problems on other class of intersection graphs.

**Acknowledgement.** Financial support offered by the Department of Science and Technology, New Delhi, India (Ref. No.SR/S4/MS/655/10) is thankfully acknowledged.

**REFERENCES**


1. A. A. Bertossi and C. M. Pinotti, Approximate L($\delta_1, \delta_2, \ldots, \delta_t$)-coloring of trees and interval graphs, *Networks*, 49(3) (2007) 204-216
2. A. A. Bertossi and M. A. Bonuccelli, Code Assignment for Hidden Terminal Interference Avoidance in Multihope Packet Radio Networks, *IEEE/ACM Transactions on Networking*, 3(4) (1995) 441-449.
3. H. L. Bodlaender, T. Kloks, R. B. Tan and J. V. Leeuwen, Approximations for $\lambda$-colorings of graphs, *The Computer Journal*, 47(2) (2004) 193-204.
4. G. J. Chang and C. Lu, Distance two labelling of graphs, *European Journal of Combinatorics*, 24 (2003) 53-58.
5. S. H. Chiang and J. H. Yan, On L(d,1)-labeling of cartesian product of a path, *Discrete Applied Mathematics*, 156 (15) (2008) 2867-2881.
6. M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs, 2ed,* Elsevier, 2004.
7. J. Griggs and R. K. Yeh, Labeling graphs with a condition at distance two, *SIAM J. Discrete Math.*, 5 (1992) 586-595.
8. W. K. Hael, Frequency Assignment: Theory and Applications, *Proc. IEEE*, 68 (1980) 1497-1514.
9. X. T. Jin and R. K. Yeh, Graph distance-dependent labelling related to code assignment in compute networks, *Naval Research Logistics*, 51 (2004) 159-164.
10. N. Khan, M. Pal and A. Pal, $L(0,1)$-lavelling of cactus graphs, *Communications and Network*, 4 (2012) 18-29.
11. S. Olariu, An optimal greedy heuristic to color interval graphs, *Information Processing Letters*, 37 (1991) 21-25.
12. A. Pal and M. Pal, Interval tree and its applications, *Advanced Modeling and Optimization*, 11 (2009) 211-224.
13. A. Pal, M. Pal and T. K. Pal, Selection of programme slots of television channels for giving advertisement: A graph theoretic approach, *Information Science*, 177(12) (2007) 2480-2492.
14. M. Pal, *Some sequential and parallel algorithms on interval graphs*, Ph.D Thesis, Indian Institute of Technology, Kharagpur, India, 1995.
15. M. Pal and G. P. Bhattacharjee, A data structure on interval graphs and its applications, *J. Circuits, Systems, and Computer*, 7 (1997) 165-175.
16. M. Pal and G. P. Bhattacharjee, Optimal sequential and parallel algorithms for computing the diameter and the center of an interval graph, *International Journal of Computer Mathematics*, 59 (1995) 1-13.
17. M. Pal and G. P. Bhattacharjee, An optimal parallel algorithm to color an interval graph, *Parallel Processing Letters*, 6 (1996) 439-449.
18. D. Sakai, Labeling chordal graphs with a condition at distance two, *SIAM J. Discrete Math.*, 7 (1994) 133-140.