TWMS J. App. Eng. Math. V.13, N.2, 2023, pp. 635-648

# A RESTRICTED $L(2,1)$-LABELLING PROBLEM ON INTERVAL GRAPHS

N. PATRA[1], SK AMANATHULLA[2], M. PAL[3*], S. MONDAL[1], §

Abstract. In a graph $G = (V, E)$, $L(2, 1)$-labelling is considered by a function $\ell$ whose domain is $V$ and codomain is set of non-negative integers with a condition that the vertices which are adjacent assign labels whose difference is at least two and the vertices whose distance is two, assign distinct labels. The difference between maximum and minimum labels among all possible labels is denoted by $\lambda_{2,1}(G)$. This paper contains a variant of $L(2, 1)$-labelling problem. In $L(2, 1)$-labelling problem, all the vertices are $L(2, 1)$-labeled by least number of labels. In this paper, maximum allowable label $K$ is given. The problem is: $L(2, 1)$-label the vertices of $G$ by using the labels $\{0, 1, 2, \dots, K\}$ such that maximum number of vertices get label. If $K$ labels are adequate for labelling all the vertices of the graph then all vertices get label, otherwise some vertices remains unlabeled. An algorithm is designed to solve this problem. The algorithm is also illustrated by examples. Also, an algorithm is designed to test whether an interval graph is no hole label or not for the purpose of $L(2, 1)$-labelling.

Keyword: Interval graph, graph labelling, $L(2, 1)$-labelling, holes in label.

AMS Subject Classification: 05C40, 05C62.

## 1. Introduction

In the mathematical graph theory, graph labeling is one of the most important problems and it has many applications to solve varieties of real life problems. The graph labelling is an allocation of labels, generally these are integers, to vertices and/or edges of a graph. On the other way, a labelling of a graph $G = (V, E)$ is a mapping $\ell$ from the set $U$ into the set of non-negative integers under certain condition(s). The set $U$ may be the set of edges or set of vertices or both. Graph labelling is one of the fascinating areas of graph theory which has a wide ranging applications. Different types of graph LPs such as simple

[1] Department of Mathematics, Raja N.L. Khan Women's College, Gope Palace, Midnapore-721102, India.
e-mail: nupurpatra93@gmail.com; ORCID: https://orcid.org/0000-0002-2556-2216.
e-mail: smnlkhan@gmail.com; ORCID: https://orcid.org/0000-0002-5639-8439.
[2] Department of Mathematics, Raghunathpur College, Raghunathpur, Purulia, 723121, India.
e-mail: amanat87math@gmail.com; ORCID: https://orcid.org/0000-0002-1281-4349.
[3] Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore 721102, India.
e-mail: mmpalvu@gmail.com; ORCID: https://orcid.org/0000-0002-6709-836X.
[*] Corresponding author.
§ Manuscript received: March 24, 2021; accepted: June 23, 2021.

| Abbreviation | Description |
|---|---|
| InvG | Interval graph |
| 2SS | 2-stable set |
| 2SSS | 2-stable subset |
| M2SS | maximal 2-stable set |
| M2SSS | maximal 2-stable subset |
| LP | labelling problem |

Table 1. A list of abbreviation

vertex labelling, edge labeling, $L(h,k)$-labeling, harmonic labeling, graceful labeling, magic labeling, anti-magic labeling, etc. are studied by many researchers. The graph LP has been applied to solve many real life problems such as scheduling, traffic planing, job assignment, etc. Particularly, labelling of interval graph (InvG) has many applications and one of them is discussed thoroughly in [33, 44]. Apart from this application, InvG is applied to solve other several problems [31, 32, 33, 34, 35, 36, 37].

There are various types of applications $L(h,k)$-labelling and so many conditions are studying in their LP. The $L(h,k)$-labelling is now studied by a huge number of researchers due to its large applications. The $L(h,k)$-LP is originated from the frequency allocation problem. Various types of frequency assignment problem was introduced by Roberts [41]. The 'very closed' transmitter has taken a frequency at least two apart and 'closed' transmitter has taken different frequency. The 'closed', 'very closed', etc. are linguistic terms and have different meaning for different persons. The assignment of frequency to a given group of televisions or radio transmitters maintaining the above conditions is called frequency assignment problem. In [26], Hael modelled this problem as vertex coloring problem.

Representing the graph of their problem, vertices of the graph are chareterised by the transmitters. Any two vertices $y$ and $z$ in a graph is called to be 'very close' if $d(y,z)=1$ and 'close' if $d(y,z)=2$, where $d(y,z)$ represents the distance between the vertices $y$ and $z$, which is the the minimum number of edges on the path connecting $y$ and $z$. In $L(h,k)$-labeling, it is generally assumed that two vertices are closed if their distance is 2 units and very closed if the distance is 1 unit. In 1992, Griggs and Yeh [25] formulated the $L(h,k)$-LP, stated below.

The $L(2,1)$-labelling of a graph $G=(V,E)$ is a function $\ell$ from $V$ to the set of non-negative integers, i.e. $\{0,1,2,\ldots\}$, such that

$$|\ell(y)-\ell(z)|\geq 2, \text{ if the distance between } y \text{ and } z \text{ is 1 in } G, \text{ and}$$
$$|\ell(y)-\ell(z)|\geq 1, \text{ if the distance between } y \text{ and } z \text{ is 2 in } G.$$

The general $L(h,k)$-LP is defined as:

$$|\ell(y)-\ell(z)|\geq h, \text{ if } y \text{ and } z \text{ are adjacent in } G \text{ and}$$
$$|\ell(y)-\ell(z)|\geq k, \text{ if } y \text{ and } z \text{ are at distance two in } G.$$

For a given graph many different labelling functions may occur. It is obvious that the domains of these functions are same which is $V$, but the co-domain may be different, even their cardinality may different. The set of such labelling functions is denoted by $\mathfrak{F}$. The function with least co-domain (least cardinality) is useful for labelling. The difference between minimum and maximum labels used to label the graph is called span.

The minimum span over all possible labelling functions $\ell\in\mathfrak{F}$ is denoted by $\lambda_{h,k}(G)$ and it is known as $\lambda_{h,k}$-number of $G$. The main objective of the problem $L(h,k)$-labelling is to minimize the span.

It can be verified that for any subgraph $H$ of a graph $G$, $\lambda_{h,k}(H) \leq \lambda_{h,k}(G)$ for $h \geq k$. The result need not be true for $h < k$. For example, let $G = K_{n+1}$ and $H = K_{1,n}$. In this case, $\lambda_{0,1}(H) = n+1$ and $\lambda_{0,1}(G) = 0$. However, the result is true, if $H$ is induced subgraph for every $h, k$.

Lot of variations of $L(h,k)$-LPs are present in literature [2, 3, 4, 5, 6, 14, 38, 39, 40, 45, 46]. One of them is $K$-$L(h,k)$-LP stated below:

Let $K$ be a given positive integer. Then $K$-$L(h,k)$-labelling of a graph $G$ is a function $\ell : V \to \{0, 1, 2, ..., K\}$ such that
$|\ell(y) - \ell(z)| \geq h$ if $d(y,z) = 1$,
$|\ell(y) - \ell(z)| \geq k$ if $d(y,z) = 2$ and
$|V'|$ is maximum, where $V'$ is the set of labeled vertices under the labelling function $\ell$.

That is in $K$-$L(h,k)$-LP a pre-specified number $K$ is given. The problem is to label the graph using $L(h,k)$-labelling approach so that the maximum label used is $K$. If the label $K$ is sufficient to label the graph using $L$-$(h,k)$-labeling, then $K$-$L(h,k)$-labeling problem is same as usual $L(h,k)$-labeling. If $K$ labels are not enough to label all the vertices of the graph then a new algorithm is required. As per our knowledge, no such algorithm is available for $K$-$L(h,k)$-labeling problem for InvG even for given $h$ and $k$.

Different bounds for $\lambda_{2,1}(G)$ are available for some special type of graphs. Let $\Delta(G)$ be the maximum among the degrees of the vertices of the graph $G$. This $\Delta(G)$ sometimes known as degree of the graph and simply denoted by $\Delta$. The size of the maximum clique is denoted by $\omega(G)$ and the chromatic number of $G$ is denoted by $\chi(G)$.

The parameters $\Delta$, $\omega(G)$ and $\chi(G)$ are used to represents the lower and upper bounds of $\lambda_{2,1}(G)$. It can easily be proved that the trivial lower bounds for $\lambda_{2,1}(G)$ are $2(\omega - 1)$ and $\Delta + 1$. In [25], Griggs and Yeh first provided the upper bound of $\lambda_{2,1}(G)$ and they shown that $\lambda_{2,1}(G) \leq \Delta^2 + 2\Delta$ for every graph $G$. This bound was improved to $\lambda_{2,1}(G) \leq \Delta^2 + \Delta$ [20]. Král' and Škrekovski [27] proved that $\lambda_{2,1}(G) \leq \Delta^2 + \Delta - 1$. This bound is further improved to $\lambda_{2,1}(G) \leq \Delta^2 + \Delta - 2$ by Goncalvas [24]. Grigges and Yeh [25] stated the following conjecture.

**Conjecture.** For any graph $G$, $\lambda_{2,1}(G) \leq \Delta^2$.

This remain an open problem. But, it is true for some specific graphs. For example, the conjecture is true for chordal graph [43] and also for InvG as it is a subclass of chordal graph.

Yeh [48] has presented following two interesting results:
(a) For any graph $G$ and the positive integer $q$, $\lambda_{qh,qk} = q\lambda_{h,k}$ ;
(b) For any graph $G$ having at least one edge,

$$\lim_{h \to \infty} \frac{\lambda_{h+1,1}(G)}{\lambda_{h,1}(G)} = 1.$$

Recently, Amanathulla *et al.* have studied lot of results regarding labeling of various types graphs [7, 8, 9, 10, 11, 12, 13, 15]. An advanced label research on graph theory is going on by Muhiuddin *et al.* [29, 30].
**Motivation:** Actually, $L(h,k)$-labelling labels all the vertices of the graph using minimum number of vertices. There is no limit about the upper bound of the label. But, the general concept is that the number of labels must be minimum. What happens if the number of available labels is less than the actual labels? In this case, some vertices must be unlabeled. But, it is not a good process to keep some vertices unlabeled. So, objective of the problem is to label the maximum number of vertices of the graph using the given labels such that $L(h,k)$-labelling condition must be satisfied.

At first Chang and Kuo [20] shown that $\lambda_{2,1}$ labelling of strongly chordal graph is at most $2\Delta$. Basically, InvG and unit InvG are nothing but a kind of strongly chordal graph. For unit InvG $G$, Sakai proved that $2\chi - 2 \leq \lambda_{2,1}(G) \leq 2\chi$ [43], where $\chi$ represents the chromatic number. Calamoneri et al. [18] proved that for an InvG the upper bound for $\lambda_{h,k}$ is $\max(h, 2k)\Delta$. When $k = 1, h = 2$, this result is coincide with the result of Chang and Kuo. Calamoneri et al. [18] also proved that $\lambda_{h,k}(G) \leq \max(h, 2k)\Delta + h\omega$ for circular arc graph. For planar graph, the decision version of $L(0, 1)$-LP is NP-complete [20].

An exhaustive survey on $L(h, k)$-LP is available in [19]. The $n$-dimensional hypercube $Q_n$ which is an $n$-regular graph having $2^n$ vertices. Then $\lambda_{0,1}(Q_n) \leq 2^{[\log n]}$ [50]. A labelling scheme is also presented for such a number of labels. When $n$ is of the form $2^t$ for some integer $t$, the this labelling is optimal and otherwise it is a 2-approximation [50]. A different approach is used in [22]. Here, an algorithm is presented that uses $2^{[\log n]+1}$ labels and the time and space complexities are $O(n)$. This improves the previous result. In both papers, the upper bound on $\lambda_{1,1}$ for $(Q_n)$ is a 2-approximation. For a bipartite graph, $\lambda_{0,1}(G) \geq \Delta^2/4$ in [17]. In [1], this lower bound is improved by a constant factor of 1/4 . The $L(d, 1)$-labelling of Cartesian product of cycles and path is investigated by Chiang and Yan [21]. This problem was proposed by Griggs and Yeh [25, 49] in connection with the problem of assigning frequency in a multiple radio network.

Rest of this paper is arranged as follows. Few important properties of InvGs are presented in Section 2. A polynomial time algorithm for $K$-$L(2, 1)$-labelling of InvG is designed in Section 3. Also, some results which are required to prove the correctness of the algorithm are presented in this section. The time complexity and correctness of the algorithm are also discussed here. In Section 4, a new upper bound of $L(2, 1)$-labelling of InvG is presented. The algorithm is illustrated in this section. In Section 5, another algorithm is designed for an InvG to test whether a hole is presented in the labelling or not. Lastly, a conclusion is drawn for the proposed work.

## 2. InvG and its properties

One of the important graphs with huge applications is InvG. A subclass of intersection graph of family of set of intervals on real line.

Let $I = \{I_1, I_2, \ldots, I_n\}$ be a set of $n$ distinct closed intervals on a real line $R$. A graph (undirected) $G = (V, E)$ is said to be an InvG if there is a bijection from $V$ to $I$. From this set of intervals $I$ one can construct a graph as follows:
For each intervals $I_j$ we consider a vertex $z_j$. And two vertices $z_i$ and $z_j$ are adjacent if and only if $I_i \cap I_j \neq \emptyset$. The graph $G = (V, E)$ constructed by this way is named as InvG and $I$ is called intersection model. It is observed that an interval and a vertex are one and same think. Note that an InvG having $n$ vertices can be stored into computer by a set of $n$ intervals. Again, one can store a set of $n$ intervals by $2n$ endpoints, i.e. using $O(n)$ space.

The set of intervals can be ordered according to the right endpoints or left endpoints, by preserving the structure of the InvG. In this paper, we assumed that the intervals are ordered according to their right endpoints.

The InvG satisfied a very nice property stated below:

**Property 1.** [23] *For an InvG $G = (V, E)$, let $z_i, z_j, z_k$ be three arbitrary vertices. If $z_i < z_j < z_k$ and $(z_i, z_k) \in E$, then $(z_j, z_k) \in E$.*

From this property, one can say that the vertices of an InvG can be ordered. A set of vertices $C$ is called a clique if all vertices of $C$ are pairwise adjacent. A clique $C$ is called maximal if $C \cup \{z_i\}$ is not a clique for any vertex $z_i \in V$.

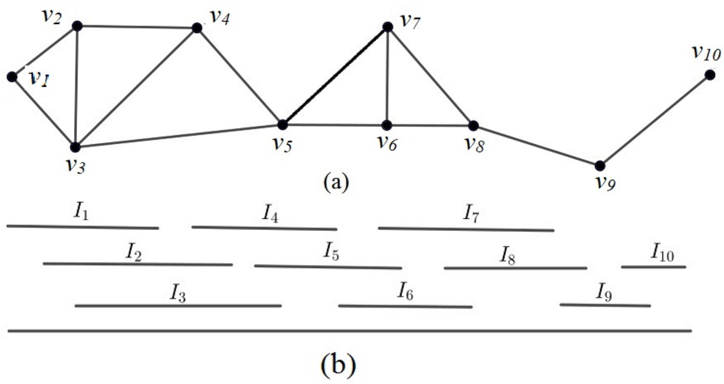


FIGURE 1. (a) An InvG; (b) Its interval representation

The following property on maximal clique is very useful to investigate the InvG.

**Property 2.** [23] *The maximal cliques of an InvG $G$ can be arranged such that for every vertex $z \in G$, the maximal cliques containing $z$ occurs consecutively.*

If the maximal cliques $C_1, C_2, \ldots, C_k$ of the InvG $G$ and $y \in C_i$ and $C_j$, $i < j$, then $y \in C_p$ for all $p$, where $i < p < j$. Again, if $y \notin C_{j+1}$, then $y \notin C_p$, for any $p > j$.

To illustrate our problem, we take into account the InvG of Figure 1.

## 3. AN ALGORITHM

A subset $S$ of the vertex set $V(G)$ is said to be a $r$-independent set or $r$-stable set, where $r$ is an integer, if $d(y,z) > r$ for any two vertices $y, z$ of $S$.

A maximal $r$-stable set $S$ of the set $F \subseteq V$ is called $s$-stable subset of $F$ if $S$ is not a proper subset of any other $s$-stable subset of $G$ contained in $F$.

Some algorithms are available to find independent set and $r$-independent set for InvGs [47]. The time complexity for finding maximal $r$-independent set of an InvG having $n$ vertices is $\mathrm{O}(n^r)$. For $r = 2$, the time complexity becomes $O(n^2)$.

The basic idea of proposed algorithm is given below:
The algorithm is repeated for $K$ times. A maximal 2-stable set (M2SS) is computed in each step among the unlabeled vertices if the distance is at least two which are labeled in the previous step. Then label all the vertices in the 2SS with the integer $r$, the index of the present step. The initial value of $r$ is 0 and $r$ is increased by 1 in every step. The final value of $r$ gives the maximum label which is needed to label all the vertices. Now, if $R$ labels be required to label a graph by $L(2,1)$-labelling and if $K$ be the allowable highest label, where $K < R$, then some vertices remains unlabeled. Let $V_u$ and $V_l$ be the sets of unlabeled and labeled vertices respectively, then $V_l = V - V_u$.

The following algorithm is designed to label all the vertices of an InvG using $K$ labels.

**Algorithm KL21**
**Input:** An InvG $G$, a non-negative integer.
**Output:** Set of labeled vertices, $V_l$.
Initially, $V_u = V$, set of unlabeled vertices. $S_{-1} = \emptyset$, $r$ is taken as 0.
**Step 1:** If $S_{r-1} = \emptyset$ then
$F_r = V_u$
else
$F_r = \{y \in V_u : d(y,z) \geq 2 \text{ and } y \text{ is unlabeled for all } z \in S_{r-1}\}$

If $F_r \neq \emptyset$ then
find $S_r$, which is the M2SSS of $F_r$
else
set $S_r = \emptyset$.
**Step 2:** Assign $r$ as label to all vertices of $S_r$, i.e. $\ell(y) = r$, for all $y \in S_r$
**Step 3:** $V_u = V_u - S_r$, as all the vertices of $S_r$ are labeled.
**Step 4:** If $V_u \neq \emptyset$ and $r < K$ then set $r = r + 1$ and go to Step 1.
**Step 5:** Repeat the above steps until $r = K$ or $V_u \neq \emptyset$.
**Step 6:** $V_l = V - V_u$, the set of labeled vertices.
**end KL21**

Note that all $S_r$, $r = 1, 2, ...$ are mutually disjoint and all $S_r$ constitute the vertex set $V$, i.e. $S_r \cap S_j = \emptyset$ and $\bigcup_r S_r = V$.

3.0.1. *An illustration.* Let us consider the InvG of Figure 1 to demonstration the algorithm KL21.

In this graph, the vertices are $V = \{v_1, v_2, ..., v_{10}\}$. Let $\ell(v_j)$ be the label assigned to the vertex $v_j \in V$ for $j = 1, 2, ..., 10$. Here, we assume that $K = 4$, the maximum allowable label. Initially, $S_{-1} = \emptyset$, $r = 0$.
**Iteration 1**: $S_{-1} = \emptyset$. So, $F_0 = V_u = V = \{v_1, v_2, ..., v_{10}\}$.
Since $F_0 \neq \emptyset$, so $S_0 = \{v_1, v_6, v_{10}\}$.
So, $\ell(1) = 0$, $\ell(6) = 0$, $\ell(6) = 0$.
$V_u = V_u - S_0 = \{v_2, v_3, v_4, v_5, v_7, v_8, v_9\}$.
Since $V_u \neq \emptyset$ and $r = 0 < 4$, so $r = r + 1 = 1$.
**Iteration 2:** $S_0 \neq \emptyset$, so $F_1 = \{v_4\}$.
Since $F_1 \neq \emptyset$, so $S_1 = \{v_4\}$.
$\therefore \ell(4) = 1$
$V_u = V_u - S_1 = \{v_2, v_3, v_5, v_7, v_8, v_9\}$.
Since $V_u \neq \emptyset$ and $r = 1 < 4$, so $r = r + 1 = 2$.
**Iteration 3:** $S_1 = \{v_4\} \neq \emptyset$, so $F_2 = \{v_7, v_8, v_9\}$.
Since, $F_2 \neq \emptyset$ therefore $S_2 = \{v_7\}$.
$\therefore \ell(7) = 2$.
$V_u = V_u - S_2 = \{v_2, v_3, v_5, v_8, v_9\}$.
**Iteration 4:** $S_2 \neq \emptyset$, so $F_3 = \{v_2, v_3, v_9\}$
Since $F_3 \neq \emptyset$, so $S_3 = \{v_2, v_9\}$.
$\therefore \ell(2) = 3$, $\ell(9) = 3$.
$\therefore V_u = V_u - S_3 = \{v_3, v_5, v_8\}$.
**Iteration 5:** $S_3 \neq \emptyset$, so $F_4 = \{v_5\}$.
Since $F_4 \neq \emptyset$, so $S_4 = \{v_5\}$.
$\therefore \ell(5) = 4$.
$\therefore V_u = V_u - S_4 = \{v_3, v_8\}$.
Here, $r = 4 = K$, so the process is terminated.

In this example, we assumed that $K = 4$ and four consecutive labels are used. The set of labeled vertices is $V_l = \{v_1, v_2, v_4, v_5, v_6, v_7, v_9, v_{10}\}$ and that of unlabeled vertices is $\{v_3, v_8\}$. The set of labels used is $\{0, 1, 2, 3, 4\}$. It can be proved that $|V_l|$ is maximum for $K = 4$.

Some useful results which required to prove the correctness and the complexity of the algorithm are discussed below.

**Lemma 1.** *Let $y$ and $z$ be any two vertices of the M2SS $S_r$ for some $r$, then $d(y, z) \geq 3$.*

**Proof.** From the definition of maximal $r$-stable set, we know that the distance between any two vertices of the set is strictly greater than $r$. Here the set $S_r$ is a M2SS. So, the distance between any two vertices of the set $S_r$ is strictly greater than 2. Here $y$ and $z$ are any two vertices of the set $S_r$. Then, from the definition, we get $d(y,z)$, i.e. the distance between the vertices $y$ and $z$ is strictly greater than 2. Therefore, $d(y,z) \geq 3$. Hence the result. □

Since the distance between any two vertices of $S_r$ is 3 or more, so the labels of the vertices of $S_r$ may be same which is done in the algorithm.

**Lemma 2.** *Suppose $y,z$ be any two vertices of the M2SSs $S_r$ and $S_{r+1}$ respectively, then $d(y,z) \geq 2$.*

**Proof.** Here, $z$ be a vertex of $S_{r+1}$. But, from the definition of $F_{r+1}$, it is also a vertex of $F_{r+1}$. If we take the vertex $y$ from $S_r$ then $d(y,z) \geq 2$ (by the definition of $F_{r+1}$).

So, if we take one vertex from the set $S_r$ and another from $S_{r+1}$, then always we get, that the distance between any two vertices one from $S_r$ and another from $S_{r+1}$ is greater than or equal to 2, i.e. $d(y,z) \geq 2$. □

In this case, it is observed that if $y \in S_r$ and $z \in S_{r+1}$, then their distance is two or more. So, the label difference between $y$ and $z$ must be at least one. In algorithm KL21, we assign $r$ to all vertices of $S_r$ and $r+1$ to all vertices of $S_{r+1}$.

**Lemma 3.** *Let $y \in S_r$ and $z \in S_{r+2}$. Then there may be an edge between the vertices $y$ and $z$, i.e. $d(y,z) \geq 1$.*

**Proof.** The distance between any two vertices of $S_r$ and $S_{r+2}$ is one or more.

Let $y \in S_r$, $z \in S_{r+1}$ and $w \in S_{r+2}$. From Lemma 2, $d(y,z) \geq 2$ and $d(z,w) \geq 2$. So, it is obvious that $d(y,z) > 1$. But, we have to prove that there may be an edge between $y$ and $w$. This can easily be proved by considering the graph of Figure 1. In this example, $S_0 = \{v_1, v_6, v_{10}\}$ and $S_2 = \{v_7\}$. From the graph it is seen that $d(v_6, v_7) = 1$ and $d(v_1, v_7) = 3 > 1$. Hence, in general, $d(v_6, v_7) \geq 1$. □

**Theorem 1.** *Algorithm L21 labels all the vertices of the InvG correctly.*

**Proof.** According to our algorithm, initially all vertices are taken as unlabeled.
Also, it is assumed that $S_{-1} = \emptyset$.

When $S_{r-1}$ is computed and all vertices of $G$ are not labeled, then we determine the set $F_r = \{y \in V\colon\ y \text{ is not labeled and } d(y,z) \geq 2 \text{ for all } z \in S_{r-1}\}$.

Next, we find a M2SSS $S_r$ of $F_r$, i.e. $S_r$ is a 2SSS of $F_r$, but $S_r$ is not a proper subset of any 2SSS of $F_r$.

It is noted that, when $F_r = \emptyset$, i.e. for any unlabeled vertex $y$ there exists a vertex $b \in S_{r-1}$ such that $d(y,z) < 2$, $S_r = \emptyset$. The distance among the vertices in $S_r$ is greater than 2 (Lemma 1). So, we can assign same label $r$ to all the vertices of $S_r$. Again, if $y \in S_r$ and $z \in S_{r+1}$, then $d(y,z) \geq 2$ (Lemma 2). In this case, one can assign label $r$ to $y$ and $(r+1)$ to $z$ as their label difference is at least one. Recall that $S_r$'s are all distinct. So, in any case, labels of the vertices in $S_r$ by the label $r$, obviously satisfy $L(2,1)$-labelling condition. □

**Theorem 2.** *The running time of Algorithm KL21 is $O(Kn^2)$, where $n$ represents the number of vertices of the graph and $K$ is an integer, the maximal allowable label.*

**Proof.** In Step 1, M2SSS of an InvG is computed and it takes $O(n^2)$ time [47]. All other assignments of this step takes time not more than $O(n)$. Since $|S_r| \leq n$ for all $r$ and $\cup S_r = V$, so to compute all $S_r$ $(r = 1, 2, \ldots)$, $O(n)$ time is required, i.e. Step 2 takes $O(n)$ time. It is obvious that Step 3 can be computed in $O(n)$ time.

Thus, the time taken by the steps 1 to 4 for a fixed $r$ is $O(n^2)$. These steps are repeated for $K$ times. Hence, the overall running time of Algorithm KL21 is $KO(n^2)$, i.e. $O(Kn^2)$, where $K$ is a given integer, the maximum allowable label. $\square$

## 4. Bound of $L(2,1)$-labelling of InvG

For some particular type of graphs such as paths, cycle, etc. the exact value of $\lambda_{2,1}(G)$ is known. But, for InvG $G$ it is very tough to find out the exact value of $\lambda_{2,1}(G)$. Now, we present an algorithm from which an upper bound of $\lambda_{2,1}(G)$ for an InvG can be determined.

The following algorithm helps us to find the upper bound of $\lambda_{2,1}(G)$ of InvG $G$.

**Algorithm L21**
**Input:** An InvG $G = (V, E)$.
**Output:** $L(2,1)$-label of each vertex of $G$.
Initialization: $S_{-1} = \emptyset$, $H = \emptyset$, $r = 0$.
**Step 1:** If $S_{-1} = \emptyset$ then $F_r = V$
else $F_r = \{y \in V : d(y, z) \geq 2$ and $y$ is unlabeled for all $z \in S_{r-1}\}$.
If $F_r \neq \emptyset$ then compute $S_r$ (M2SSS of $F_r$)
else set $S_r = \emptyset$.
**Step 2:** Assign $r$ as the label to all vertices of $S_r$, i.e. $\ell(z) = r$, for all $z \in S_r$.
**Step 3:** $V = V - S_r$.
**Step 4:** If $V$ is non-empty then set $r = r + 1$ and then go to Step 1.
**Step 5:** Repeat above steps until $V = \emptyset$.
Set $K = r$ (last label used)
Stop
End.

Let $z$ be a vertex whose label is $K$, i.e. $\ell(z) = K$. Now, we define three sets of vertices below.

Let
$J_1(x) = \{z : d(x, y) = 1$ and $0 \leq z \leq (K - 1)$ for some $y \in S_z\}$.
$J_2(x) = \{z : d(x, y) \leq 2$ and $0 \leq z \leq (K - 1)$ for some $y \in S_z\}$.
$J_3(x) = \{z : d(x, y) \geq 3$ and $0 \leq z \leq (K - 1)$ for all $y \in S_z\}$.
That is, $J_1(y)$ is the set of labels of the neighborhood of $y$.
$J_2(y)$ is the set of labels of the vertices whose distance from $y$ is at most two.
$J_3(y)$ is the set of labels which are not used by any vertex whose distance from $y$ is at most three.

Thus, the sum of the cardinalities of $J_2(z)$ and $J_3(z)$ is $K$.
That is, $|J_2(z)| + |J_3(z)| = K$.
For any $r \in J_3(z)$, $z \notin F_r$; otherwise $S_r \cup \{z\}$ is a 2SSS of $F_r$, which contradicts the selection of $S_r$.

Thus, $d(y, z) = 1$ for some vertex $z \in S_{r-1}$.
Therefore, $r - 1 \in J_1(z)$ so that $|J_3(z)| \leq |J_1(z)|$.
Then $\lambda_{2,1}(G) \leq k = |J_2(z)| + |J_3(z)| \leq |J_2(z)| + |J_1(z)|$.

Hence, $\lambda_{2,1}(G) \leq |J_2(z)| + |J_1(z)| \quad (1)$

**Lemma 4.** *For any InvG $G$, $|N_2(z)| \leq 2\Delta$ where $N_2(z)$ is the set of vertices with distance two apart from $z$.*

**Proof.** The maximum number of vertices adjacent to the vertex $z$ is $\Delta$. Let $|N_2(z)| = s$. This implies that there are $s$ distinct vertices which are at a distance two from the vertex $z$.

Since $G$ is an InvG, so there exists two vertices which are adjacent to at most $\Delta$ vertices of $G$. Therefore, $s \leq 2\Delta$, i.e. $|N_2(z)| \leq 2\Delta$. □

Note that Algorithm L21 can label an InvG $G$ by maintaining the condition of $L$(2,1)-labelling. So, this algorithm may be used to $L$(2,1)-label an InvG $G$. For this usefulness, one can determine the time complexity of this algorithm.

Note that Algorithms KL21 and L21 both are same, only their termination conditions are different. Algorithm KL21 repeats for $K$ time whereas Algorithm L21 terminates when all the vertices become labeled. Thus by replacing $K = 4\Delta$, in Theorem 2 we get the following results.

**Theorem 3.** *The algorithm L21 labels all the vertices of an InvG $G$ having $n$ vertices using $O(\Delta n^2)$ time.*

This situation does not always happen. Some labels can not be used due to the structure of the graph. These unused labels are called hole. This case is discussed in next section.

## 5. Holes in $L(2,1)$-labelling of InvG

In this section, we discuss about holes in $L(2,1)$-labelling for InvGs and obtained a good result. The definition of hole is discussed.

**Definition 1.** *Let $l$ be an $L(2,1)$-labelling of a graph $G$ that uses labels from $0$ to $\lambda$. Then an integer $p$ is called a hole, if $p \in (0, \lambda)$ and there exists no vertex $z \in V$ such that $\ell(z) = p$. The maximum number of holes in a span L(2,1)-labelling of a graph $G$ is denoted by $H_\lambda(G)$.*

The holes can be identified by $L$(2,1)-labelling of InvG using Algorithm $L21$. Suppose $L = \{0, \ldots, \lambda\}$ be the set of labels used to $L(2,1)$-label. The labels 0 and $\lambda$ are used surely. If an $r$, $0 < r < \lambda$ does not belong to $L$ then this $r$ is a hole. So, by checking this condition for $r = 1, 2, \ldots, \lambda - 1$ one can determine the set of holes $H_\lambda(G)$ for a graph $G$.

Following, we designed an independent algorithm which will determine the set of holes and also test whether a given InvG is 'no hole $L(2,1)$-label'.

**Algorithm HL21**
**Input:** An InvG $G = (V, E)$.
**Output:** Holes $H$ in $L$(2,1)-labelling.
Initialization: $S_{-1} = \emptyset, r = 0$, $H = \emptyset$.
**Step 1:** If $S_{-1} = \emptyset$ then $F_r = V$
else $F_r = \{z \in V(G) : z$ is not labeled and $d(z, y) \geq 2$ for all $y \in S_r\}$.
If $F_r \neq \emptyset$ then compute $S_r$ (M2SSS of $F_r$)
else set $S_r = \emptyset$.
**Step 2:** If $S_r = \emptyset$ then $H = H \cup \{r\}$
else $V = V - S_r$.
**Step 3:** If $V \neq \emptyset$, then set $r = r + 1$ and go to step 1.
**Step 4:** Repeat the above steps until $V = \emptyset$.
**Step 5:** If $H = \emptyset$ then $G$ is 'no hole $L$(2,1)-labeled'
otherwise $G$ has hole.
End.

Let us take the InvG for Figure 2 in which some labels are not used, i.e. there is hole. In this graph $V = \{v_1, v_2, ..., v_8\}$.

Thus, $S_{-1} = \emptyset$, $r = 0$.

**Iteration 1:** $S_{-1} = \emptyset$, so $F_0 = V_u = V = \{v_1, v_2, ..., v_8\}$.
Since $F_0 \neq \emptyset$, so $S_0 = \{v_1, v_4, v_6\}$.

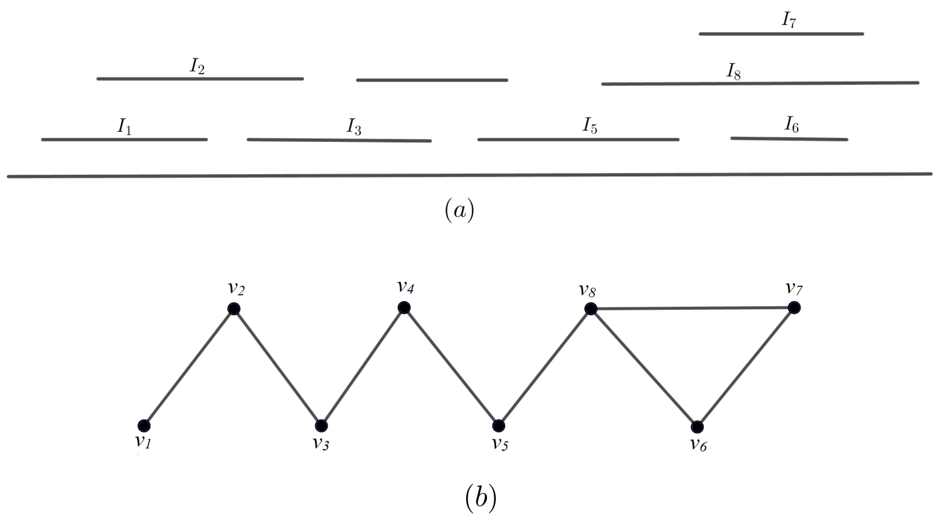


FIGURE 2. Illustration of hole of $L(2,1)$-labelling for InvG

So, $\ell(1) = 0$, $\ell(4) = 0$, $\ell(6) = 0$.
$V_u = V_u - S_0 = \{v_2, v_3, v_5, v_7, v_8\}$.
Since $V_u \neq \emptyset$ so $r = r + 1 = 1$.
**Iteration 2:** $S_0 \neq \emptyset$, so $F_1 = \emptyset$ and hence $S_1 = \emptyset$.
Since $S_1$ is empty so label 1 can not be used to label any vertex.
$V_u = V_u - S_1 = \{v_2, v_3, v_5, v_7, v_8\}$.
Since $V_u \neq \emptyset$ so $r = r + 1 = 2$.
**Iteration 3:** $S_1 = \emptyset$, so $F_2 = \{v_2, v_3, v_5, v_7, v_8\}$.
Since $F_2 \neq \emptyset$, so $S_2 = \{v_2, v_5\}$.
Therefore, $\ell(2) = 2$, $\ell(5) = 2$.
$V_u = V_u - S_2 = \{v_3, v_7, v_8\}$ and $r = 3$.
**Iteration 4:** $S_2 \neq \emptyset$, so $F_3 = \{v_7\}$.
Since $F_3 \neq \emptyset$, so $S_3 = \{v_7\}$.
$\therefore \ell(7) = 3$.
$V_u = V_u - S_3 = \{v_3, v_8\}$.
Since $V_u \neq \emptyset$ so $r = r + 1 = 4$.
**Iteration 5:** $S_3 \neq \emptyset$, so $F_4 = \{v_3\}$.
$F_4 \neq \emptyset$, so $S_4 = \{v_3\}$.
$\therefore \ell(3) = 4$.
$V_u = V_u - S_4 = \{v_8\}$.
**Iteration 6:** In this stage, there is only one vertex is left to label and it is 8. It can be verified that label 5 can be used to label the vertex 8.

The labels and the set of vertices are shown in Table 2. For labelling the graph we used the labels $\{0, 2, 3, 4, 5\}$ of Figure 2.

This example shows that the labels used to $L$(2,1)-label the graph are not consecutive integers. Here one label remains unused. So, the set of holes for this graph is $\{1\}$.

The time complexity to find holes of an InvG in case of $L(2,1)$-labelling is stated below:

| Label | Set of vertices |
|---|---|
| 0 | $\{1,4,6\}$ |
| 2 | $\{2,5\}$ |
| 3 | $\{7\}$ |
| 4 | $\{3\}$ |
| 5 | $\{8\}$ |

Table 2. Labels and corresponding set of vertices

**Theorem 4.** *The running time to find the set of holes $H_\lambda(G)$ for an InvG $G$ in case of $L(2,1)$-labelling is $O(\Delta n^2)$, where $n$ and $\Delta$ represent the number of vertices and degree of the graph respectively.*

## 6. Concluding remarks

In this paper, $K$-$L(2,1)$-LP has been considered. A polynomial time algorithm is designed to solve this problem on InvGs. Also, it is shown that the upper bound to $L(2,1)$-label an InvG is $4\Delta$ which is much better that the conjuncture for $L(2,1)$-LP. Again, no hole problem is discussed and presented an algorithm to test whether an InvG has no hole or not in case of $L(2,1)$-labelling. The no hole problem for $L(2,1)$-labelling is studied for few classes of graph specially intersection graphs. So, our algorithm can be extended for other subclass of intersection graph particularly for circular-arc graph.

**Acknowledgement.** The authors are thankful to the reviewers for their valuable comments for the better presentation of the work.

Financial support of first and fourth author offered by DHESTBT (Govt. of West Bengal, India) (Ref. No. 245(Sanc.)/ST/P/S&T/16G-20/2017) is thankfully acknowledged.

## References

[1] Alon,N., Mohar, B., (2002), The chromatic number of graph powers, Combinatorics, Probability and Computing, 11(1), 1-10.

[2] Amanathulla, S., Pal, M., (2016), $L(0,1)$- and $L(1,1)$-labeling problems on circular-arc graphs, International Journal of Soft Computing, 11(6), 343-350.

[3] Amanathulla, S., Pal, M., (2016), $L(3,2,1)$- and $L(4,3,2,1)$-labeling problems on circular-arc graphs, International Journal of Control Theory and Applications, 9(34), 869-884.

[4] Amanathulla, S., Pal, M., (2017), $L(h_1,h_2,\dots,h_m)$-labeling problems on interval graphs, International Journal of Control Theory and Applications, 10(1), 467-479.

[5] Amanathulla, S., Pal, M., (2017), $L(3,2,1)$-labeling problems on permutation graphs, Transylvanian Review, 25(14), 3939-3953.

[6] Amanathulla, S., Pal, M., (2017), $L(3,2,1)$- and $L(4,3,2,1)$-labeling problems on interval graphs, AKCE International Journal of Graphs and Combinatorics, 205-215.

[7] Amanathulla, S., Pal, M., (2018), Surjective L (2, 1)-labeling of cycles and circular-arc graphs, Journal of Intelligent and Fuzzy Systems, 35 (1). 739-748.

[8] Amanathulla, S., Sahoo, S., Pal, M., (2018), $L(3,1,1)$-labeling numbers of square of paths, complete graphs and complete bipartite graphs, Journal of Intelligent and Fuzzy Systems, 36, 1917-1925.

[9] Amanathulla, S., Pal, M., (2018), $L(1,1,1)$- and $L(1,1,1,1)$-labeling problems of square of paths, complete graphs and complete bipartite graphs, Far East Journal of Mathematical Sciences, 106(2), 515-527.

[10] Amanathulla, S., Pal, M., (2020), $L(3,2,1)$-labeling problems on trapezoid graphs, Discrete Mathematics Algorithms and Applications, DOI: 10.1142/S1793830921500683.

[11] Amanathulla, S., Bera, B., Pal, M., (2021), Real world applications of discrete mathematics, Malaya Journal of Mathematik, 9(1), 152-158.

[12] Amanathulla, S., Bera, B., Pal, M., (2021), surjective $L(3,1)$-labeling of circular-arc graphs, Malaya Journal of Mathematik, 9(1), 925-929.

[13] Amanathulla, S., Bera, B., Pal, M., (2021), $L(2,1,1)$-labeling of circular-arc graphs, Journal of Scientific Research, 13(2), 537-544.

[14] Amanathulla, S., Pal, M., (2017), $L(h_1,h_2,\dots,h_m)$-labeling problems on circular-arc graphs, Far East Journal of Mathematical Sciences, 102(6), 1279-1300.

[15] Amanathulla, S. Muhiuddin, G., Al-Kadi, D., Pal, M., (2021), Distance two surjective labelling of paths and interval graphs, Discrete Dynamics in Nature and Society ,doi.org/10.1155/2021/9958077.

[16] Bertossi, A., Pinotti, C., (2007) Approximate L($\delta_1,\delta_2,\dots,\delta_t$)-coloring of trees and interval graphs, Networks, 49(3), 204-216.

[17] Bodlaender, H, Kloks, T,. Tan, R., and Leeuwen, J., (2004), Approximations for $\lambda$-colorings of graphs, The Computer Journal, 47(2), 193-204.

[18] Calamoneri, T., Caminiti, S., Olariu, S., Petreschi, R., (2009), On the $L(h,k)$-labelling of co-comparability graphs and circular-arc graphs, Networks, 53(1), 27-34.

[19] Calamoneri, T., (2011), The $L(h,k)$-labelling problem: an updated survey and annotated bibliography, J. Comput, 54(8), 1344-1371.

[20] Chang, G., Kuo, D., (1996), The $L(2,1)$-labelling on graphs, SIAM Journal of Discrete Mathematics, 9, 309-316.

[21] Chiang, S., Yan, J., (2008), On L(d,1)-labelling of cartesian product of a path, Discrete Applied Mathematics, 156(15), 2867-2881.

[22] Das, S., Finocci, I., Petreschi, R., (2006), Conflict-free star-access in parallel memory systems, J. Parallel Distrib. Syst., 66, 1431-1441.

[23] Golumbic, M., (2004), Algorithmic Graph Theory and Perfect Graphs, 2ed, Elsevier.

[24] Goncalves, D., (2008), On the $L(d,1)$-labellinng of graphs, Discrete Math., 308, 1405-1414.

[25] Griggs, J., Yeh, R., (1992), Labelling graphs with a condition at distance two, SIAM J. Discrete Math., 5, 586-595.

[26] Hael, W, (1980), Frequency Assignment: Theory and Applications, Proc. IEEE, 68, 1497-1514.

[27] Král', D., Škrekovski, R., (2003), A theory about channel assignment problem, SIAM J. Discrete Math., 16, 426-437.

[28] Olariu, S., (1991), An optimal greedy heuristic to color interval graphs, Information Processing Letters, 37, 21-25.

[29] Muhiuddin, G., Sridharan, N., Al-Kadi, D., Amutha, S., Elnair1, M., (2021), Reinforcement number of a graph with respect to half-domination, Journal of Mathematics, , doi.org/10.1155/2021/6689816.

[30] Muhiuddin, G., Alanazi, A, Kannan, A., Govindan, V., (2021), Preservation of the classical meanness property of some graphs based on line graph operation, Journal of Mathematics, doi.org/10.1155/2021/4068265.

[31] Pal, M., (1995), Some sequential and parallel algorithms on interval graphs, Ph.D Thesis, Indian Institute of Technology, Kharagpur, India, 1995.

[32] Pal, M., Bhattacharjee, G., (1995), Optimal sequential and parallel algorithms for computing the diameter and the center of an interval graph, International Journal of Computer Mathematics, 59, 1-13.

[33] Pal, M., Bhattacharjee, G., (1996), An optimal parallel algorithm to color an interval graph, Parallel Processing Letters, 6, 439-449.

[34] Pal, M., Bhattacharjee, G., (1996), A sequential algorithm for finding a maximum weight k-independent set on interval graphs, International Journal of Computer Mathematics, 60(3-4), 205-214.

[35] Pal, M., Bhattacharjee, G., (1997), A data structure on interval graphs and its applications, J. Circuits, Systems, and Computer, 7, 165-175.

[36] Pal, M., Bhattacharjee, G., (1997), An optimal parallel algorithm for all-pairs shortest paths on unweighted interval graphs, Nordic Journal of Computing, 4(4), 342-356.

[37] Pal, M., (2013), Intersection graphs: an introduction, Annals of Pure and Applied Mathematics, 4(1), 41-93.

[38] Paul, S., Pal, M., Pal, A., (2014), $L(2,1)$-labelling of circular-arc graph, Annals of Pure and Applied Mathematics, 5(2), 208-219.

[39] Paul, S., Pal, M., Pal, A., (2015), $L(2,1)$-labelling of interval graph, Journal of Applied Mathematics and Computing, 49(1), 419-432.

[40] Paul, S., Pal, M., Pal, A., (2016), A linear time algorithm to compute square of interval graphs and their colouring, AKCE International Journal of Graphs and Combinatorics, 13(1), 54-64.

[41] Roberts, F., (1991), $T$-coloring of graphs: recent results and open problem, Discrete Math., 93, 229-245

[42] Rosa, A., (1967), On certain valuations of vertices in a graph International Symposium Rome, Italy.

[43] Sakai, D., (1994), Labelling chordal graphs with a condition at distance two, SIAM J. Discrete Math., 7, 133-140.

[44] Saha, A., Pal, M., and Pal, T., (2007), Selection of programme slots of television channels for giving advertisement: A graph theoretic approach, Information Science, 177(12), 2480-2492.

[45] Shao, Z., Yeh, R., (2007), The $L(2,1)$-labelling on planar graphs, Appl. Math. Lett., 20, 222-226.

[46] Shao, Z., Yeh, R., Zhang, D., (2008), The $L(2,1)$-labelling on graphs and the frequency assignment problem, Applied Mathematics Letters, 21(1), 37-41.

[47] Hsio, J., Tang, C, Chang, R., (1992), An efficient algorithm for finding a maximum weight 2-independent set on interval graphs, information Processing Letters, 43(5), 229-235.

[48] Yeh, R., (2006), A survey on labelling graphs with a condition at distance two, Discrete Mathematics, 306, 1217-131.

[49] Yea, R., (1990), Labeling graphs with condition at distance two, Ph.D Thesis, University of South Carolina, Columbia.

[50] Wan, P., (1997), Near-Optimal Conflit-free channel set assignment for an optical cluster-based hypercube network, J.Combin.Optim., 1, 179-186.

[51] Nayeem, S., Pal, M., (2013), Diameter constrained fuzzy minimum spanning tree problem, International journal of computational intelligence systems, 6(6), 1040-1051.

[52] Pal, M., Samanta, S., Ghorai, G., (2020), Modern Trends in Fuzzy Graph Theory, Springer Singapore.

[53] Barman, S., Mondal, S., Pal, M., (2013), Minimum 2-tuple dominating set of permutation graphs, Journal of Applied Mathematics and Computing 43(1), 133-150.

[54] Bera, D., Pal, T, Pal, M., (1999), An optimal parallel algorithm to compute all cutvertices and blocks on permutation graphs, International journal of computer mathematics, 72(4), 449-462.

[55] Barman, S., Pal, M., Mondal, S., (2010), The k-neighbourhood-covering problem on interval graphs, International Journal of Computer Mathematics, 87(9), 1918-1935.

[56] Barman, S., Mondal, S., Pal, M., (2010), A linear time algorithm to construct a tree 4-spanner on trapezoid graphs, International Journal of Computer Mathematics, 87(4), 743-755.

[57] Rana, A., Pal, A., Pal, M., (2011), The 2-neighbourhood covering problem on permutation graphs, Advanced Modeling and Optimization 13(3), 463-476.

**Nupur Patra** is currently a state aided college teacher of Mathematics at Raja N. L. Khan Women's College. She completed her M.Sc in Applied Mathematics in 2017 at Vidyasagar University and currently she is pursuing her Ph.D. Her specialization is algorithmic and fuzzy graph theory.

**Dr. Sk Amanathulla** is currently an assistant professor of mathematics at Raghunathpur College, Raghunathpur. He completed his M.Sc and Ph.D from Vidyasagar University. His specializations includes algorithmic and fuzzy graph theory.

**M. Pal** for the photography and short autobiography, see TWMS J. App. and Eng. Math. V.8, N.2.

**Dr. Sukumar Mondal** is currently an associate professor of Mathematics at Raja N. L. Khan Women's College. His research interest on Graph Theory and Fuzzy Graph Theory.