TWMS J. App. and Eng. Math. V.14, N.3, 2024, pp. 1254-1263

$L(2,1)$-LABELING OF TRAPEZOID GRAPHS

S. PAUL[1], SK AMANATHULLA[2*], M. PAL[1], A. PAL[3], §

Abstract. An $L(2,1)-$*labeling* ($L21L$) of a graph $G = (V, E)$ is an assignment f from the node-set V to the set $\{0, 1, 2, 3, \ldots\}$ so that adjoining nodes get numbers at least two apart, and nodes at distance two get different numbers. The $L21L$ number $\lambda_{2,1}(G)$ is the difference between the greatest and least label used in the labeling process. In this paper, we have proved that, for a trapezoid graph (TG) G, the upper bound of $\lambda_{2,1}(G) \leq 5\Delta - 4$, where Δ is the maximum degree of the graph G. This paper also provides $L21L$ of a simple triangle graph, a subclass of TG. We have shown that for a simple triangle graph, the upper bound of $\lambda_{2,1}(G)$ is 4Δ.

Keyword: Frequency assignment, L(2,1)-labeling, trapezoid graphs.

AMS Subject Classification: 05C85, 68R10.

1. Introduction

$L21L$ problem was introduced by Griggs and Yeh [21], initially proposed by Roberts [29], as a variation of a frequency assignment problem. The definition of $L21L$ of a graph is given below.

Definition 1.1. *For any graph $G = (V, E)$, $L21L$ of is an assignment $f : V \to \{0, 1, 2, \ldots, \lambda\}$ such that $|f(u) - f(v)| \geq 2$ if $d(u, v) = 1$ and $|f(u) - f(v)| \geq 1$ if $d(u, v) = 2$, λ being a suitable integer.*

The span of $L21L$ f of G is the difference between the greatest and the least used labels. $\lambda_{2,1}(G)$ is the minimum span over all possible labeling functions.

Griggs and Yeh [21] first showed that $\lambda_{2,1}(G) \leq \Delta^2 + 2\Delta$ for any graph G. It was improved by Chang and Kuo [17] to $\lambda_{2,1}(G) \leq \Delta^2 + \Delta$. Král' *et al.* [24] showed that $\lambda_{2,1}(G) \leq \Delta^2 + \Delta - 1$. Later, Gonçalves [20] improved it to $\lambda_{2,1}(G) \leq \Delta^2 + \Delta - 2$. Griggs

[1] Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapore 721102, India.
e-mail: satyamath09@gmail.com, ORCID: https://orcid.org/0000-0002-6709-836X.
mmpalvu@gmail.com; ORCID: https://orcid.org/0000-0003-1635-8253.
[2] Department of Mathematics, Raghunathpur College, Raghunathpur 723121, India.
e-mail: amanat87math@gmail.com; ORCID: https://orcid.org/0000-0002-7313-6107.
[*] Corresponding author.
[3] Department of Mathematics ,National Institute of Technology Durgapur, Durgapur 713209, India.
e-mail: anita.buie@gmail.com; ORCID: https://orcid.org/0000-0002-2514-5463
§ Manuscript received: October 10, 2022; accepted: December 28, 2022.
TWMS Journal of Applied and Engineering Mathematics, Vol.14, No.3,

and Yeh [21] conjectured that for any graph G, $\lambda_{2,1}(G) \leq \Delta^2$. Havet et al. [23] have proved this result asymptotically. This conjecture is true for several graph classes, such as paths, cycles, wheels [21], trees [17, 22], cographs [17], chordal graphs [30], permutation graphs [15, 27], bipartite permutation graphs [1, 27], interval graphs [17, 25, 28], etc. To know more about this problem, see some comprehensive surveys [16, 31] of this field.

Recently, we have investigated the problem on various types of intersection graphs, such as circular-arc graphs [4, 5, 8, 10, 13, 14, 26], interval graphs [7, 27], permutation graphs [6, 9, 28], etc. We feel that the problem may be NP-complete for the above class of graphs. Thus, we have found some upper bounds of $L21L$ of these classes of graphs. TGs is another class of intersection graphs and it is the generalization of interval and permutation graphs. No bound is known for $L21L$ ofTGs and simple triangle graphs. Also, there are no algorithms designed for $L21L$ of these graphs. Motivated by these, we have studied the $L21L$ problem of TGs. Obviously, the problem is more complex than interval and permutation graphs. Recently, some authors are trying work on fuzzy colouring [2, 3]. In future we will tyr to fuzzyfi $L(2,1)$-labeling of some classes of graphs.

The remaining part of the paper is organized as follows. Section 2 contains some preliminary results and some notations. Section 3 contains the $L21L$ algorithm of TG and the proof of the upper bound. Section 4 discusses the $L21L$ algorithm of simple triangle graphs. Finally, in the last section, conclusions are made.

2. Preliminaries and notations

The graphs studied in this article are simple. The distance between the nodes u and v, denoted by $d(u,v)$ is the shortest distance between u and v. The neighbourhood of a node u is $N_G(u) = \{v|uv \in E\}$. The set $N_G(u)$ is also known as 1-nbd. node set of u or $N_1(u)$. Similarly, 2-nbd. nodes of u is defined as $N_2(u) = \{v|d(u,v) = 2\}$. $d(v_i)$ represents the degree of the node v_i. The maximum degree among all the nodes of the graphs is called the degree of a graph G, denoted by Δ or $\Delta(G)$.

Definition 2.1. *A TG consists of two horizontal lines L_1 (upper line) and L_2 (lower line) and a set of trapezoids $T = \{T_1, T_2, \ldots, T_n\}$ with corner points lying on these two lines. A graph $G = (V,E)$ is a TG, when a trapezoid diagram exists with trapezoid set T, such that each node $v_i \in V$ corresponds to a trapezoid $T_i \in T$ and an edge $(v_i, v_j) \in E$ if and only if $T_i \cap T_j \neq \emptyset$.*

A trapezoid T_i between these two lines has four corner points a_i, b_i, c_i, and d_i which represent the upper left, upper right, lower left and lower right corner points respectively. Without loose any generality, we assume that no two trapezoids share a common endpoint. The terms node and trapezoid are used interchangeably whenever the context is unambiguous. In this paper, it is assumed that a trapezoid diagram is given and the trapezoids are labeled in increasing order of their upper left corner points, that is, $T_i < T_j$ or $v_i < v_j$ if and only if $a_i < a_j$. Figure 1 shows an trapezoid representation and its corresponding TG.

The class of TGs is a very important subclass of the intersection graph. It was introduced by Dagan et al. [19].

2.1. Notation.

Some notations are presented in this part which we use later to develope the article.

For each trapezoid $T_j, j = 1, 2, \ldots, n$, we define a^j, b^j, c^j and d^j as follows.

$a^j = \{\min a_i : T_i \cap T_j \neq \phi \text{ and } a_i < a_j\}$. If no such trapezoid T_i exist then $a^j = 0$.

$b^j = \{\max b_i : T_i \cap T_j \neq \phi \text{ and } b_i > b_j\}$. If no such trapezoid T_i exist then $b^j = 0$.

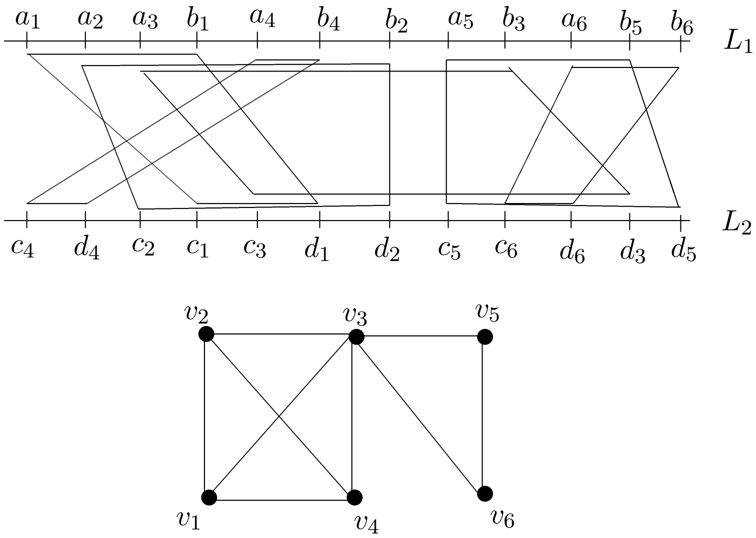

FIGURE 1. A trapezoid representation and the corresponding TG G

$c^j = \{\min c_i : T_i \cap T_j \neq \phi$ and $c_i < c_j\}$. If no such trapezoid T_i exist then $c^j = 0$.
$d^j = \{\max d_i : T_i \cap T_j \neq \phi$ and $d_i > d_j\}$. If no such trapezoid T_i exist then $d^j = 0$.

From the above notations, it is clear that a^j and c^j lies on L_1. Similarly, b^j and d^j lies on L_2. By using the above notation, we prove the following lemmas.

Lemma 2.1. *Let T_j be a trapezoid corresponding to the node v_j. If a^j is the a's corner point of the trapezoid T_{i_1} and c^j is the c's corner point of the trapezoid T_{i_2}, then $d(v_{i_1}, v_{i_2}) = 1$.*

Proof. From the definition of a^j and c^j, it is clear that v_{i_1} and v_{i_2} are adjacent to v_j.

a^j is the upper left corner of the trapezoid T_{i_1}. That is $a^j = a_{i_1}$ and it is the minimum among all upper left corners of the trapezoids which are adjacent to T_j. So, obviously $a_{i_1} < a_{i_2}$.

Again c^j is the c's corner point of the trapezoid T_{i_2}. Thus $c^j = c_{i_2}$ and it is the minimum among all lower left corners of the trapezoids which are adjacent to T_j. So, obviously $c_{i_2} < c_{i_1}$.

From the above two relations we have the line segments joining the points a_{i_1}, c_{i_1} and a_{i_2}, c_{i_2} cut each other. That is, the trapezoid T_{i_1} and T_{i_2} have non-empty intersection. Therefore $d(v_{i_1}, v_{i_2}) = 1$. Hence the result. □

Lemma 2.2. *Let T_j be the trapezoid corresponding to the node v_j. If b^j is the b's corner point of the trapezoid T_{i_1} and d^j is the d's corner point of the trapezoid T_{i_2}, then $d(v_{i_1}, v_{i_2}) = 1$.*

Proof. The proof is similar to Lemma 2.1. □

3. L(2,1)-labeling of trapezoid graphs

In this portion, an algorithm to L(2,1)-label a TG has been presented. Based on this algorithm we found an upper bound of $L21L$ for TG. Here, we prove that the upper bound of $L21L$ for TG is $5\Delta - 4$.

Consider the node set $V = \{v_1, v_2, \ldots, v_n\}$, where $v_1 < v_2 < \ldots < v_n$ and $1, 2, \ldots, n$ are the indices of corresponding nodes. We denote f_i, the label of the node v_i, for all

$i = 1, 2, \ldots, n$. We define two sets A_i and B_i below.
$A_i = \{v_j : d(v_j, v_i) = 1 \text{ and } j < i\}$ and
$B_i = \{v_j : d(v_j, v_i) = 2 \text{ and } j < i\}$,

Lemma 3.1. *For a node $v_i \in V$, $|B_i| \leq 2\Delta - 4$.*

Proof. Let T_i be the trapezoid corresponding to the node v_i. Now we calculate the cardinality of the set B_i.

Let v_{i_1} and v_{i_2} be two adjacent nodes of v_i such that $a_{i_1} = a^i$ and $c_{i_2} = c^i$. That is, a^i is the a's corner point of the trapezoid T_{i_1} corresponding to the node v_{i_1}. Similarly, c^i is the c's corner point of the trapezoid T_{i_2} corresponding to the node v_{i_2}. So, all the 2-nbd.. nodes of v_i with index less than i (i.e. the members of the set B_i) must be adjacent to either v_{i_1} or v_{i_2} or both.

Again from Lemma 2.1, v_{i_1} and v_{i_2} are adjacent. So v_{i_1} is adjacent to at most $\Delta - 2$ nodes except the nodes v_i and v_{i_2}. Similarly, v_{i_2} is also adjacent to at most $\Delta - 2$ nodes except the nodes v_i and v_{i_1}. Thus the cardinality of the set B_i is at most $2(\Delta - 2)$ i.e. $2\Delta - 4$. □

Now we design a greedy algorithm to label all the nodes of a TG.

Algorithm TGL21

Input: Set of ordered nodes of TG.
//assume that the trapezoid are ordered according to the upper left corner points. That is, $a_i < a_j$ for $v_i < v_j$ and hence $v_1 < v_2 < \ldots < v_n$.//
Output: f_i, the $L(2,1)$-label of v_i, $i = 1, 2, \ldots, n$.
for each $i = 1$ to n do
    Let j be the smallest non-negative integer such that
    $j \notin \{f_k - 1, f_k, f_k + 1 | v_k \in A_i\} \cup \{f_l | v_l \in B_i\}$
    $f_i = j$;
end for;
end TGL21

Lemma 3.2. *Algorithm TGL21 correctly $L(2,1)$-label a TG.*

Proof. From Algorithm TGL21, it is clear that the labels of each node v_i depend only on the labels of the nodes of A_i and B_i. Let $v_k \in A_i$ and $v_l \in B_i$. From Algorithm TGL21, it follows that f_i forbid the integers $f_k - 1, f_k, f_k + 1$ and f_l. Therefore, obviously, $|f_k - f_i| \geq 2$ and $|f_l - f_i| \geq 1$, where f_k and f_l are the labels of the nodes v_k and v_l respectively. That is, the label of every node follows the $L21L$ condition. Therefore, the nodes are correctly labeled. □

Theorem 3.1. *The running time for the algorithm TGL21 is $O(m + n\Delta)$.*

Proof. here, f_i, the label of v_i depends only on the labels of the sets A_i and B_i. Now A_i can be found by calculating the adjacency of the node v_i. So, $O(\sum d(v_i))$, i.e. $O(m)$ time are required to compute all A_i's. Again, let T_{i_1} and T_{i_2} be two trapezoids such that $a_{i_1} = a^i$ and $c_{i_2} = c^i$. Thus, from Lemma 3.1, all the members of the set B_i are adjacent to either v_{i_1} or v_{i_2} or both. So, all the B_i's can be computed in $O(\sum d(v_{i_1}) + \sum d(v_{i_2}))$ time, which is at most $O(2 \sum \Delta)$, i.e. $O(n\Delta)$. Hence, the overall time complexity of algorithm TGL21 is $O(m + n\Delta)$. □

In the next theorem, the upper bound of $L21L$ is stated.

Theorem 3.2. *For a TG G, $\lambda_{2,1}(G) \leq 5\Delta - 4$.*

Proof. Suppose, we are in the midst of the algorithm and going to label the ith node, i.e. v_i. Clearly, the label of the node v_i depends only on the set A_i and B_i as we label the nodes of the graph by $L21L$. Thus the total number of forbidden colors for the node v_i is at most $3|A_i| + |B_i|$ as each 1-nbd.. node of v_i forbid at most 3-integers and each 2-nbd.. node forbids at most 1-integer. Again A_i is the set of 1-nbd. nodes of v_i with index less than i. So, obviously $|A_i| \leq \Delta$, for all i. Again, from Lemma 3.1, $|B_i| \leq 2\Delta - 4$. Thus,

$$\begin{array}{rrcl} & f_i & \leq & 3|A_i| + |B_i| \\ or, & f_i & \leq & 3\Delta + 2\Delta - 4 \\ or, & f_i & \leq & 5\Delta - 4 \end{array}$$

Hence $\lambda_{2,1}(G) \leq 5\Delta - 4$. □

4. $L21L$ of simple triangle graphs

A simple triangle graph is a very important subclass of TG where the objects are triangles with one endpoint of the triangle on L_1 and the other two points (i.e. interval) of the triangle on L_2. This graph is also known as a point-interval graph or PI graph. This class was introduced by Corneil et al. [18].

Now we design an interval representation of a simple triangle graph and deduce some important properties using this interval representation.

4.1. **Interval representation of a simple triangle graph.** Let $A = \{A_1, A_2, \ldots, A_n\}$ be the set of triangles between two horizontal lines L_1 and L_2. In a simple triangle graph, each triangle A_i contain three points (a_i, b_i, c_i), where the interval $[a_i, b_i]$ (say I_i) lies on the line L_2 and the point c_i lies on the line L_1. We assume that the nodes of a simple triangle graph are ordered by increasing the left endpoint of the intervals, i.e. $v_i < v_j$ if and only if $a_i < a_j$.

Now we consider lower line (L_2) and upper line (L_1) as u and v axis in R^2. For each A_i, $i = 1, 2, \ldots, n$, we draw a line segment joining the point $(a_i, 0)$ and $(b_i, 0)$ at c_i. That is, the line segment $I_i = [a_i, b_i]$ lies on the straight line $y = c_i$. Figure 2 shows an interval representation of a simple triangle graph.

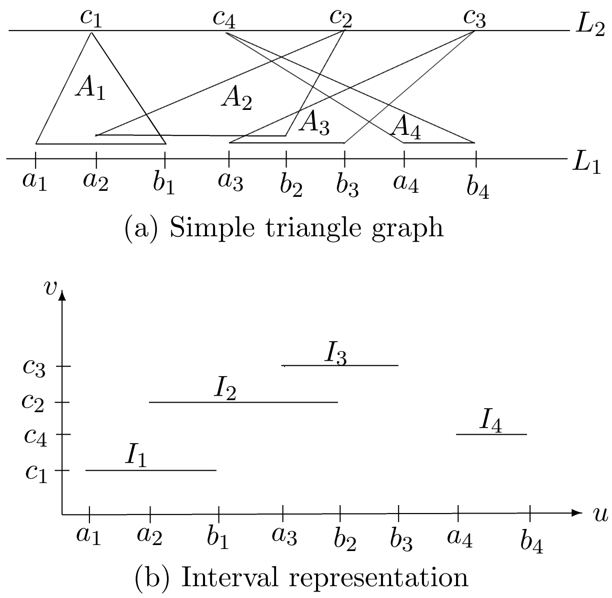

FIGURE 2. Interval representation of a simple triangle graph

From the above interval representation of a simple triangle graph, we conclude the following results.

Lemma 4.1. *If two intervals $I_i = [a_i, b_i]$ and $I_j = [a_j, b_j]$ have a non-empty intersection then $d(v_i, v_j) = 1$.*

Proof. Let $I_i = [a_i, b_i]$ and $I_j = [a_j, b_j]$ be two intervals in the interval representation of a simple triangle graph. Now, if two intervals have a non-empty intersection then the triangles T_i and T_j corresponding to the intervals I_i and I_j must intersect each other. Hence $d(v_i, v_j) = 1$. □

Lemma 4.2. *Let $I_i = [a_i, b_i]$ and $I_j = [a_j, b_j]$ be two intervals corresponding to v_i and v_j, such that $I_i < I_j$ (i.e. $a_i < a_j$) and $c_i > c_j$, then $d(v_i, v_j) = 1$.*

Proof. If I_i and I_j have a non-empty intersection then obviously $d(v_i, v_j) = 1$.

Suppose $I_i \cap I_j = \phi$ and $c_i > c_j$. That is $a_i < a_j$ and $c_i > c_j$. So the line segments $[a_i, c_j]$ and $[a_j, c_j]$ must intersect each other. Thus $T_i \cap T_j \neq \phi$. Hence the result. □

From the above two lemmas, it is clear that the interval representation of a simple triangle graph is not equivalent to an interval graph because the adjacency of an interval graph can be tasted by using only Lemma 4.1, but for simple triangle graph, both the above two lemmas are true. So, we conclude that, for every interval graph, there exists a simple triangle graph but the converse is not true. Actually, an interval graph is a subclass of TG as well as a simple triangle graph.

Let v_i, v_j and v_l be three nodes such that $v_i < v_j < v_l$ (i.e. $a_i < a_j < a_l$) then we call the node v_j as intermediate node of v_i and v_l.

Lemma 4.3. *If $d(v_i, v_j) = 1$ then the number of nodes between v_i and v_j including v_i and v_j is at most 2Δ.*

Proof. Two cases aries.
Case 1. When $a_i < a_j$ and $c_i < c_j$ (see Figure 3)

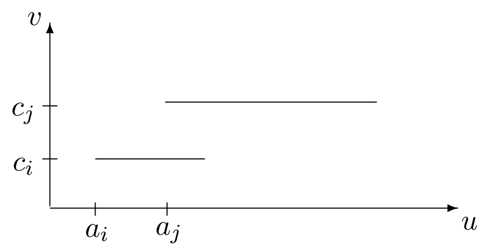

FIGURE 3. When $a_i < a_j$ and $c_i < c_j$

Since degree of the graph is Δ, v_i is adjacent to at most Δ nodes. So, in this case the number of intermediate nodes between v_i and v_j is at most $\Delta - 1$.
Case 2. When $a_i < a_j$ and $c_i > c_j$ (see Figure 4)

From Lemma 4.2, it is clearly $d(v_i, v_j) = 1$. Now, the degree of the graph is Δ. So, between a_i and b_i, there exist at most $\Delta - 1$ lower left endpoints as v_j is adjacent to v_i but $a_j \notin [a_i, b_i]$. Again v_j is adjacent to at most $\Delta - 1$ nodes except the node v_i such that the left endpoints of these adjacent nodes are less than a_j. Thus, between a_i and a_j there exist at most $(\Delta - 1) + (\Delta - 1)$ i.e. $2\Delta - 2$ left endpoints.

Therefore, from the above two cases, we conclude that the number of nodes between v_i and v_j including v_i and v_j is at most $(2\Delta - 2) + 2$, i.e. 2Δ. □

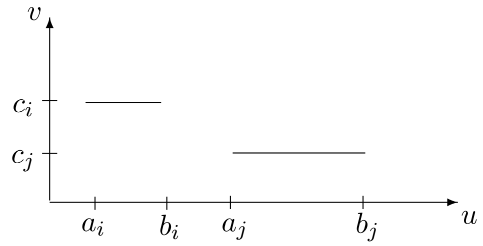

FIGURE 4. When $a_i < a_j$ and $c_i > c_j$

Lemma 4.4. *Let v_i, v_j and v_l be three nodes such that $v_i < v_j < v_l$ and $d(v_i, v_l) = 2$ through the node v_j, then the number of nodes between v_i and v_l including v_i and v_l is at most $4\Delta - 1$.*

Proof. Let v_i, v_j and v_l be three nodes such that $v_i < v_j < v_l$ and $d(v_i, v_l) = 2$ through the node v_j. That is $d(v_i, v_j) = 1$, $d(v_j, v_l) = 1$ and $d(v_i, v_l) = 2$. Again, from Lemma 4.3, the intermediate nodes between v_i and v_j is at most $2\Delta - 2$. Similarly, the intermediate nodes between v_j and v_l is at most $2\Delta - 2$. So the number of nodes between v_i and v_l including v_i, v_j and v_l is at most $(2\Delta - 2) + (2\Delta - 2) + 3$ i.e. $4\Delta - 1$. □

4.2. **An algorithm.** In this section, we design the $L21L$ algorithm for a simple triangle graph. The strategy of our algorithm is as follows. First, we calculate the degree of a simple triangle graph. Now, we consider a label set L, a set of non-negative integers defined below.

$$L = \{0, 2, 4, \dots, 4\Delta, 1, 3, 5, \dots, 4\Delta - 1\}.$$

Let $V = \{v_1, v_2, \dots, v_n\}$, where $v_1 < v_2 < \dots < v_n$ (i.e. $a_1 < a_2 < \dots < a_n$) be the set of ordered nodes. Now we label the ordered nodes by the labels from the label set L consecutively and when all the labels of L are used by the nodes then we reuse the labels maintaining the same order for the remaining nodes starting from 0. That is $f_1 = 0, f_2 = 2, f_3 = 4, \dots$. The process will continue until all the nodes get labeled.

A formal algorithm is given below.

Algorithm STGL21

Input: Set of ordered nodes $V = \{v_1, v_2, \dots, v_n\}$ of a simple triangle graph.
Output: f_i, the $L(2,1)$-label of v_i, $i = 1, 2, \dots, n$.
Step 1. Calculate the degree of the graph, i.e. Δ.
Step 2. Consider the label set $L = \{0, 2, 4, \dots, 4\Delta, 1, 3, 5, \dots, 4\Delta - 1\}$.
Step 3. Label the ordered nodes by the labels $0, 2, 4, \dots, 4\Delta, 1, 3, 5, \dots, 4\Delta - 1$ consecutively and for the remaining nodes repeat these labels in same order. The process will continue until all the nodes are labeled.
end STGL21

Theorem 4.1. *Time complexity of algorithm STGL21 is $O(m + n)$.*

Proof. Step 1 of algorithm STGL21 takes $O(m)$ time. That is, the label set L can be constructed in $O(m)$ time. In step 3, $O(n)$ time is required to label all the nodes. Therefore, the overall time complexity of algorithm STGL21 is $O(m + n)$. □

The correctness and upper bound of $\lambda_{2,1}(G)$ of a simple triangle graph are provided.

Theorem 4.2. *Let G be a simple triangle graph, then $\lambda_{2,1}(G) \leq 4\Delta$.*

Proof. In Algorithm STGL21, a simple triangle graph is labeled by using the label from the label set L. Now we show that our algorithm follows $L21L$ condition and the label set L is sufficient to label a simple triangle graph. Thus, our main aim is to shown that if $|f_i - f_j| = 1$ then $d(v_i, v_j) > 1$ and if $f_i = f_j$ then $d(v_i, v_j) > 2$.

Case 1. Let v_i and v_j be two nodes such that $f_i = 2k$ and $f_j = 2k-1$, where $k \in \{1, 2, \ldots, 2\Delta\}$.

From the label set L, it is clear that between $2k$ and $2k-1$, there are $2\Delta - 1$ integers in the set L. So, including $2k$ and $2k-1$, there are $2\Delta + 1 (\geq 2\Delta)$ integers. Again in Lemma 4.3, it is shown that, if $d(v_i, v_j) = 1$ then the number of nodes between v_i and v_j including $v_i$and v_j is at most 2Δ. So, in this case, $d(v_i, v_j) > 1$. That is, f_i and f_j satisfy $L21L$ condition.

Case 2. Let v_i and v_j be two nodes such that $f_i = f_j$.
From the label set L, it is clear that the same label repeats after 4Δ number of integers. That is, if $f_i = f_j$ then the number of nodes between v_i and v_j is at most 4Δ which is greater than $4\Delta - 1$. Thus, from Lemma 4.4, we conclude that $d(v_i, v_j) > 2$. Thus the proposed algorithm follows the $L21L$ condition and the label set L is enough to label a simple triangle graph. Now, $\max L = 4\Delta$. So $\lambda_{2,1}(G) \leq 4\Delta$. Hence the result. □

5. Conclusion

The $L21L$ problem has been widely studied over the last three decades. There are only a few classes of graphs for which they have efficient algorithms. For the other classes of graphs, a good upper bound is clearly welcome. Here we estimate the upper bounds of these classes of graphs. Also, polynomial time algorithms are designed to label such graphs. Unfortunately, our algorithms do not give a guarantee that the value of $\lambda_{2,1}$ is the least. Also, it is well known that the complexity of $L21L$ of interval [28] and permutation graphs [27] are still open and TG is the generalization of interval and permutation graph. So, for TGs, the complexity is still open. Thus, we conclude the paper by presenting some open problems.

Problem 1. Reduce these upper bounds for the proposed graphs.

Problem 2. Design optimal algorithms for $L21L$, or prove that the problem is NP-complete for these graphs.

Compliance with ethical standards.

Disclosure statement. No potential conflict of interest was reported by the authors.

Ethical approval. This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Araki, T., (2009), labelling bipartite permutation graphs with a condition at distance two, Discrete Applied Mathematics, 157, pp. 1677-1686.

[2] Borzooei, R. A., Rashmanlu, H., Pal, M. and Samanta, S. (2016), A study on fuzzy labeling graphs, Journal of Intelligent and Fuzzy Systems, 30(6), pp. 3349-3355.

[3] Samanta, S., Pramanik, T. and Pal, M. (2016), Fuzzy colouring of graphs, Afrika Matematika, 27(1), pp. 37-50.

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

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

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

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

[8] Amanathulla, S., Pal, M., (2017), $L(h_1, h_2, \ldots, h_m)$-labelling problems on circular-arc graphs, Far East Journal of Mathematical Sciences, 102(6), pp. 1279-1300.
[9] Amanathulla, S., Pal, M., (2017), $L(h_1, h_2, \ldots, h_m)$-labelling problems on interval graphs, International Journal of Control Theory and Applications, 10(01), pp. 467-479.
[10] Amanathulla, S., Pal, M., (2018), Surjective $L(2,1)$-labeling of cycles and circular-arc graphs, Journal of Intelligent and Fuzzy Systems, 35, pp. 739-748.
[11] Amanathulla, S., Pal, M., Sahoo, S., (2019), $L(3,1,1)$-labelling numbers of squares of paths, complete graphs and complete bipartite graphs, Journal of Intelligent and Fuzzy Systems, 36(2), pp. 1917-1925.
[12] Amanathulla, S., Pal, M., (2021), $L(3,2,1)$-labelling problems on Trapezoid graphs, Discrete Mathematics, Algorithms and Applications, doi.org/10.1142/S1793830921500683.
[13] Amanathulla, S., Bera, B., Pal, M., (2021), $L(2,1,1)$-labeling of circular-arc graphs, Journal of Scientific Research, 13(2), pp. 537-544.
[14] Amanathulla, S., Bera, B., Pal, M., (2021), Surjective $L(3,1)$-labeling of circular-arc graphs, Malaya Journal of mathematik, 9(1), pp. 925-929.
[15] Bodlaender, H.L., Kloks, T., Tan, R.B., Leeuwen, J.V., (2004), Approximations for λ-colorings of graphs, The Computer Journal, 47(2), pp. 193-204.
[16] Calamoneri, T., (2011), The $L(h,k)$-labelling problem: An updated survey and annotated bibliography, The Computer Journal, 54(8), pp. 1344-1371.
[17] Chang, G.J., Kuo, D., (1996), The $L(2,1)$-labelling on graphs, SIAM J. Discrete Math., 9, pp. 309-316.
[18] Corneil, D. G., Kamula, P. A., (1987), Extensions of permutation and interval graphs, Congr. Numer., 58, pp. 267-275.
[19] Dagan, I., Golumbic, M. C., Pinter, R. Y., (1988), Trapezoid graphs and their coloring, Discr. Appl. Math., 21, pp. 35-46.
[20] Goncalves, D., (2008), On the $L(d,1)$-labellinng of graphs, Discrete Math., 308, pp. 1405-1414.
[21] Griggs, J., Yeh, R.K., (1992), Labelling graphs with a condition at distance two, SIAM J. Discrete Math., 5, pp. 586-595.
[22] Hasunuma, T., Ishii, T., Ono, H., Uno, Y., (2009), A linear time algorithm for $L(2,1)$-labelling of trees, Lecture Notes in Computer Science, 5757, pp. 35-46.
[23] Havet, F., Reed, B., Sereni, J. S., (2008), $L(2,1)$-labelling of graphs, in: Proc. 19th annual ACM-SIAM Symposium on Discrete Algorithms, SODA, SIAM, 621-630.
[24] Král', D., Škrekovski, R., (2003), A theory about channel assignment problem, SIAM J. Discrete Math., 16, pp. 426-437.
[25] Paul, S., Pal, M., Pal, A., (2013), An efficient algorithm to solve $L(0,1)$-labelling problem on interval graphs, Advanced Modeling and Optimization, 15(1), pp. 31-43.
[26] Paul, S., Pal, M., Pal, A., (2014), $L(2,1)$-labelling of circular-arc graph, Annals of Pure and Applied Mathematics, 5(2), pp. 208-219.
[27] Paul, S., Pal, M., Pal, A., (2014), $L(2,1)$-labelling of permutation and bipartite permutation graphs, Mathematics in Computer Science. DOI: 10.1007/s11786-014-0180-2.
[28] Paul, S., Pal, M., Pal, A., (2014), $L(2,1)$-labelling of interval graphs, J. Appl. Math. Comput., DOI 10.1007/s12190-014-0846-6.
[29] Roberts, F.S., (1991), T-Colorings of graphs: recent results and open problems, Discrete Mathematics, 93, pp. 229-245.
[30] Sakai, D., (1994), Labelling chordal graph: distance two condition, SIAM J. Discrete Math., 7, pp. 133-140.
[31] Yeh, R.K., (2006), A survey on labelling graphs with a condition at distance two, Discrete Math., 306, pp. 1217-1231.

Dr. Satyabrata Paul is currently an Assistant Teacher in Mathematics at a school. His research interests include graph theory, fuzzy graph theory, Topological indices, and chemical graphs.

Dr. Sk Amanathulla is an Assistant Professor of Mathematics at Raghunathpur College, Raghunathpur, West Bengal, India. He has completed his M. Sc and Ph. D from Vidyasagar University. His specializations include Algorithmic and Fuzzy Graph Theory, Genetic Algorithms, and Parallel Algorithms.

M. Pal for the photography and short autobiography, see TWMS J. App. and Eng. Math. V.14, N.1.

Dr. Anita Pal is an Associate Professor of Mathematics at the National Institute of Technology, Durgapur, West Bengal, India. Her specializations include Algorithmic and Fuzzy Graph Theory, Genetic Algorithms, and Parallel Algorithms.