TWMS J. App. and Eng. Math. V.15, N.6, 2025, pp. 1418-1434

A STUDY ON MAXIMUM CARDINALITY r-$L(2,1)$-LABELLING PROBLEM ON CIRCULAR-ARC GRAPH AND ITS APPLICATION

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

ABSTRACT. Graph labelling is one of the most applicable problem in graph theory, often applied to solve real-world challenges. This article explores a range of $L(2,1)$-labelling problems (L21LPs), specifically focusing on the r-L21LP within CirGs. In the standard L21LP, each vertex in a graph is assigned a label from a set of non-negative integers. The labeling follows these rules: for adjacent vertices, the label difference must be at least 2; for vertices at distance two, the label difference must be at least 1; and for vertices farther apart, there are no label restrictions. The difference between the highest and lowest labels among all vertices is denoted as $\lambda_{2,1}(G)$. This paper introduces a variation of the $L(2,1)$-labelling problem, known as the restricted L21LP, where a maximum label limit r is imposed. Consequently, the valid labels are restricted to $\{0,1,2,\ldots,r\}$. The objective is to $L(2,1)$-label the vertices of G using these limited labels to maximize the number of labelled vertices. If the available r labels suffice to label all vertices, then every vertex is labelled; otherwise, some vertices remain unlabelled. A polynomial-time algorithm is proposed to address this problem, along with illustrative examples. Additionally, an application scenario is presented, demonstrating the use of this labelling scheme to allocate program slots on telecasting channels for advertising products or disseminating information for organizations.

Keyword: Circular-arc graph; graph labelling; $L(2,1)$-labelling; design and analysis of algorithm.

AMS Subject Classification: 05C72.

---

[1]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.
[2] Department of Mathematics, Raghunathpur College, Raghunathpur, India.
e-mail: amanat87math@gmail.com; ORCID:https://orcid.org/0000-0002-7313-6107.
e-mail: smnlkhan@gmail.com; ORCID: https://orcid.org/0000-0002-5639-8439.
[3] Department of Applied Mathematics, Vidyasagar University, Midnapore 721102, India.
e-mail: mmpalvu@gmail.com; ORCID: https://orcid.org/0000-0002-6709-836X.
* Corresponding author.
§ Manuscript received: April 19, 2024; accepted: July 09, 2024.
TWMS Journal of Applied and Engineering Mathematics, Vol.15, No.6; © Işık University, Department of Mathematics, 2025; all rights reserved.
The first author is financially supported by DHESTBT (Government of West Bengal, India), under Ref. No245(Sanc.)/ST/P/S&T/16G-20/2017.

Abbreviation	Description
InvG	Interval graph
CirG	Circular-arc graph
L21L	L(2,1)-labelling
L21LP	L(2,1)-labelling problem
LhkL	L(h,k)-labelling
LhkLP	L(h,k)-labelling problem

Table 1. Table of Abbreviation

In this article we use some abbreviation given in Table 1.

1. Introduction

In mathematical graph theory, graph labelling stands out as a fundamental problem with wide-ranging applications in solving real-world challenges. It involves assigning labels, typically integers, to the vertices and/or edges of a graph. More formally, a graph labelling for a graph $G = (V, E)$ is a mapping ℓ from a set U into the set of non-negative integers, subject to certain conditions. The set U may represent the set of vertices, edges or both. This area of graph theory is both fascinating and highly applicable, offering solutions to a diverse array of problems. The intersection graphs are very useful to model different types of real world problems [6, 8, 18, 37, 38, 36, 46].

Various types of graph labelling problems, such as simple vertex labelling, edge labelling, $L(h, k)$-labelling, harmonic labelling, graceful labelling, magic labelling, anti-magic labelling, etc., have been extensively studied by researchers. These labelling problems find applications in scheduling, traffic planning, job assignment, and more. In particular, the labelling of interval graphs (InvG) and circular-arc graphs (CirG) has been explored extensively, showcasing their relevance in practical scenarios.

Moreover, InvG and CirG have been instrumental in solving numerous other problems beyond graph labelling, as evidenced by their applications in various contexts [31, 32, 45, 46]. These graphs have proven to be versatile tools in addressing a wide range of challenges, further underscoring the significance of graph theory in problem-solving.

It is true that the real world is full of uncertainties, which are addressed by probability theory or fuzzy theory. Consequently, fuzzy graph theory has been developed, and extensive research has been conducted in this field. Mondal et al. [40, 41] have recently investigated the utilization of m-polar fuzzy graph to address road network issues, employing the isometric and antipodal concepts and generalized m-polar fuzzy planar graph. Some of the very novel work are vailable in [10, 7, 12, 24, 25, 27, 28, 29, 39, 42].

The $L(h, k)$-labelling problem (LhkLP) has found numerous applications and has been extensively studied by researchers due to its wide-ranging practical implications. Its origins trace back to the frequency allocation problem, which was introduced by Roberts [43]. In this context, various types of frequency assignment problems have been investigated [2, 3, 4, 33, 34, 35, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 26].

In the frequency allocation problem, different linguistic terms such as 'very closed' and 'closed' are used to describe the relationship between transmitters and their assigned frequencies. Specifically, a 'very closed' transmitter is allocated a frequency that is at least two units apart from other frequencies, while a 'closed' transmitter is assigned a different frequency. However, the exact definitions of these terms may vary depending on the context.

The task of assigning frequencies to a given group of televisions or radio transmitters while adhering to the above conditions is known as the frequency assignment problem. Hale [21] modeled this problem as a vertex coloring problem.

In this modeling approach, the graph's vertices represent the transmitters, and the edges between vertices represent the relationships between transmitters. Two vertices p and q are considered 'very close' if their distance, denoted by $dist(p, q)$, is 1 unit, and 'close' if their distance is 2 units. Here, the distance between vertices is defined as the minimum number of edges on the path connecting them.

The LhkLP typically operates under the assumption that two vertices are considered 'closed' if their distance is 2 units apart, and 'very closed' if their distance is 1 unit. Griggs and Yeh [20] formalized the LhkLP, specifically defining the $L(2, 1)$-labelling (L21L) of a graph $G = (V, E)$ as a function ℓ mapping vertices in V to non-negative integers $\{0, 1, 2, \ldots\}$, satisfying the conditions:

$$|\ell(p) - \ell(q)| \geq 2, \text{ if the } dist(p, q) \text{ is 1 in } G, \text{ and}$$
$$|\ell(p) - \ell(q)| \geq 1, \text{ if the } dist(p, q) \text{ is 2 in } G.$$

The general LhkLP is defined as follows:

$$|\ell(p) - \ell(q)| \geq h, \text{ if } dist(p, q) \text{ is 1 in } G \text{ and}$$
$$|\ell(p) - \ell(q)| \geq k, \text{ if } dist(p, q) \text{ is 2 in } G.$$

The LhkLP has attracted considerable attention from researchers because of its diverse practical applications. Its origins can be traced back to Roberts' introduction of the frequency allocation problem [43]. This problem encompasses various types of frequency assignment scenarios, where different linguistic terms like 'very closed' and 'closed' are employed to describe the relationship between transmitters and their allocated frequencies. Specifically, a 'very closed' transmitter is assigned a frequency that is at least two units apart from other frequencies, while a 'closed' transmitter is given a different frequency. However, the exact definitions of these terms may vary depending on the specific context.

The task of assigning frequencies to a group of radio transmitters or televisions while adhering to the aforementioned conditions is known as the frequency assignment problem. Hale [**?**] approached this problem by modeling it as a vertex coloring problem.

In this modeling approach, the graph's vertices represent the transmitters, and the edges between vertices represent the relationships between transmitters. Two vertices p and q are considered 'very close' if their distance, denoted by $dist(p, q)$, is 1 unit, and 'close' if their distance is 2 units. Here, the distance between vertices is defined as the minimum number of edges on the path connecting them.

The LhkLP typically assumes that two vertices are 'closed' if their distance is 2 units apart, and 'very close' if their distance is 1 unit. Griggs and Yeh [20] formalized the LhkLP in 1992, with the L21L of a graph $G = (V, E)$ defined as a function ℓ from V to the set of non-negative integers $\{0, 1, 2, \ldots\}$, such that:

Consider a given positive integer r. Then, an r-LhkL of a graph G is defined by a function $\ell : V \to \{0, 1, 2, ..., r\}$ such that:
(i) $|\ell(p) - \ell(q)| \geq h$ if $dist(p, q) = 1$,
(ii) $|\ell(p) - \ell(q)| \geq k$ if $dist(p, q) = 2$, and
(iii) $|V'|$ is maximum, where V' represents the set of labeled vertices under the labelling function ℓ.

CirG is a very important subclass of intersection graph. Let $A = \{A_1, A_2, \ldots, A_n\}$ be a set of n arcs around a circle. The CirG can be constructed from this set of arcs as follows: For each arc an vertex is considered. If two arcs overlap then there is an edge between the corresponding vertices and if there is no overlap, then there is no edge between the vertices.

In the context of the r-LhkLP, a predefined integer r is given, and the objective is to label the graph using the LhkL approach while ensuring that the maximum label used does not exceed r. If the label r is adequate to label the entire graph using L-(h,k)-labelling, then the r-LhkLP reduces to the standard LhkL. However, if the maximum label r falls short of labelling all vertices in the graph, a new algorithm is necessitated. Presently, no such algorithm exists for solving the r-LhkLP for circular graphs, even for given values of h and k.

Various bounds for $\lambda_{2,1}(G)$ are available for certain types of graphs. Let $\Delta(G)$ denote the maximum degree of the vertices in graph G, sometimes referred to simply as the graph's degree and denoted by Δ.

Griggs and Yeh [20] initially provided an upper bound for $\lambda_{2,1}(G)$, demonstrating that $\lambda_{2,1}(G) \leq \Delta^2 + 2\Delta$ for all graphs G. This bound was subsequently improved to $\lambda_{2,1}(G) \leq \Delta^2 + \Delta$. Gonçalves [19] made additional improvements, establishing $\lambda_{2,1}(G) \leq \Delta^2 + \Delta - 2$. Griggs and Yeh [20] also proposed the following conjecture.

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

This problem remains unresolved, although it holds true for certain particular graphs. For instance, it is true for the conjecture has been verified for chordal graphs and InvG, these are the subclass of chordal graphs.

In [61], Yeh made two noteworthy observations:

(a) For a positive integer q and any graph G, $\lambda_{qh,qk} = q\lambda_{h,k}$.

(b) For any non-null graph G,

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

Motivation. In the realm of graph theory, the LhkL problem seeks to assign labels to vertices in a graph with the minimal number of labels possible, without imposing constraints on the label values. The primary objective is to minimize the total number of labels used, ensuring that as many vertices as possible receive labels while adhering to the LhkL condition. Initially explored by Chang and Kuo [15], they established that for strongly chordal graphs, the $\lambda_{2,1}$ labeling is bounded above by 2Δ, where Δ represents the maximum degree of the graph. This finding extends to InvG and unit InvG graphs. Sakai [44] further contributed to this field by demonstrating that for unit InvG graphs, the $\lambda_{2,1}$ labeling falls within the range of $2\chi - 2 \leq \lambda_{2,1}(G) \leq 2\chi$, where χ is the chromatic number. Calamoneri et al. [13] expanded upon this research, proving that for InvG graphs, the upper bound for $\lambda_{h,k}$ is $\max(h, 2k)\Delta$. Particularly, when $k = 1$ and $h = 2$, their results coincide with those of Chang and Kuo. Additionally, Calamoneri et al. [14] demonstrated that for Circular Graphs (CirG), $\lambda_{h,k}(G) \leq \max(h, 2k)\Delta + h\omega$, where ω denotes the clique number. Notably, the decision version of the LhkLP for $h = 0, k = 1$ is NP-complete for planar graphs [15].

A comprehensive review of LhkLP is provided by Calamoneri et al. [14]. For n-dimensional hypercubes Q_n, Wan [63] established that $\lambda_{0,1}(Q_n) \leq 2^{[\log n]}$, providing a corresponding labeling scheme. This scheme proves optimal when n takes the form 2^t for some integer t, otherwise constituting a 2-approximation. Das et al. [17] proposed an alternative algorithm utilizing $2^{[\log n]+1}$ labels with time and space complexities of $O(n)$, representing an advancement in efficiency over previous methods. For bipartite graphs, Bodlaender et al. [11] established that $\lambda_{0,1}(G) \geq \Delta^2/4$, a lower bound subsequently improved by a constant factor of 1/4 in [1]. Chiang and Yan [16] explored the L(d, 1)-labeling of Cartesian products of cycles and paths, a problem initially introduced by Griggs and Yeh [20, 62] in the context of frequency assignment in multiple radio networks.

Our work. In this paper, we address the r-LhkLP and introduce a solution algorithm with a time complexity of $O(n\Delta^2)$, where n denotes the number of vertices in the graph and Δ represents the maximum degree of the graph. We offer an illustrative example to showcase the algorithm's effectiveness. To the best of our knowledge, no existing algorithm can solve the r-LhkLP for CirG, even when the parameters h and k are provided.

Additionally, we explore an application involving the selection of program slots from telecasting channels. We present an $O(n\Delta^2)$ time algorithm designed to address this problem, accompanied by a suitable example to illustrate its functionality.

The remainder of this paper is structured as follows:

Section 2: We define CirG, providing necessary background for the subsequent discussion.

Section 3: We present a polynomial time algorithm specifically designed to solve the r-L21LP for CirG. We also include essential results required to establish the correctness of the algorithm, and discuss its time complexity.

Section 4: We introduce an application scenario related to program slot selection from telecasting channels.

This paper aims to contribute to the field of graph theory by providing practical solutions to challenging labelling problems and demonstrating their real-world applicability.

2. Circular-arc graph

As we navigate clockwise around the circle, each arc is defined by its "starting point," where it's encountered initially, and its "finishing point," where it concludes. Denoted as A_i, each arc corresponds to a closed interval $[a_j, b_i]$, with a_j marking its counterclockwise end (starting point) and b_i indicating its clockwise end (finishing point), satisfying $a_j < b_i$. Dividing the arcs are lines extending from the circle's center through the finishing points, segregating them into two sets: "backward arcs" ($\mathcal{S}_B$), intersected by the line, and "forward arcs" ($\mathcal{S}_F$), untouched. If k denotes the number of arcs intersected by this line, expressed as $|\mathcal{S}_B| = k$, the backward arcs' endpoints are initially labeled clockwise from 1 to k, while the rest, from $k+1$ to n, are sequentially labeled based on their starting points. This labeling establishes an order of arcs based on increasing counterclockwise ends. When two arcs A_i and A_j share common points on the circle, they're termed "intersecting arcs"; otherwise, they're deemed "independent arcs."

The primary focus lies on scenarios where every point on the circle is covered by at least one arc. If any portion of the circle lacks coverage by any arc within a Circular Graph (CirG), it reveals a gap, simplifying the CirG's arc model to an interval model enveloping the circle. Consequently, algorithms tailored for interval graphs can then determine the maximum weight independent set.

A key aspect is the equivalence between the arcs and vertices of the CirG, which results from its construction using the set of arcs.

To ascertain adjacency between two arcs (or vertices) $A_i = [a_j, b_i]$ and $A_j = [a_k, b_k]$, a straightforward method can be employed.

Lemma **2.1.** *For any two adjacent arcs $A_i = [a_j, b_i]$ and $A_j = [a_k, b_k]$, one of the following conditions is true*

(i) $a_j < a_k < b_i < b_k$ *or*

(ii) $a_j < a_k < b_k < b_i$ *or*
(iii) $a_k < a_j < b_i < b_k$.

To exemplify our issue, we consider the CirG depicted in Figure 1.

FIGURE 1. (a) An example of circular-arc graph; (b) The circular arc representation of the graph of Fig. (a)

In the following section, an algorithm is presented to solve the r-L21LP for CirG.

3. An algorithm

Notations:
$J(A_k)$: Set of labels used before labelling arc A_k.
$J_1(A_k)$: Set of labels at distance 1 from arc A_k before labelling A_k.
$J_2(A_k)$: Set of labels at distance 2 from arc A_k before labelling A_k.
$J_{vl}(1, A_k)$: Set of valid labels for labelling arc A_k before labelling A_k, satisfying adjacency conditions.
$J_{vl}(2, A_k)$: Set of valid labels for labelling arc A_k before labelling A_k, satisfying the L21L condition.
$l(k)$: Label of arc A_k or vertex v_k.
Here we present an algorithm to compute $J_{vl}(1, A_k)$ and $J_{vl}(2, A_k)$, $k = 2, 3, \ldots, n$.

Algorithm KVL
Input: The set of arcs $A_k, k = 2, 3, \ldots, n$.
Output: $J_{vl}(r, A_k)$ for $r = 1, 2; k = 2, 3, \ldots, n$.
Step 1: Compute $J_1(A_k), J_2(A_k)$ and $J(A_k)$.
for $i = 1$ to p where $p = \max\{J(A_k)\} + 2$
for $j = 1$ to $|J_1(A_k)|$
if $|i - l_q| \geq 2$, then adding element i to the set $J_{vl}(1, A_k)$,
i.e., $J_{vl}(1, A_k) = \{i\}$, where l_q be the qth element of $J_1(A_k)$.
end for;
end for;
Step 2:
for $s = 1$ to $|J_{vl}(1, A_k)|$
for $t = 1$ to $|J_2(A_k)|$
if $|l_s - p_t| \geq 1$, then $J_{vl}(2, A_k) = \{l_s\}$,
where $l_s \in J_{vl}(1, A_k)$, $p_t \in J_2(A_k)$.
end for;
end for;
end KVL

Theorem 3.1. *The algorithm KVL correctly compute* $J_{vl}(1, A_k)$ *and* $J_{vl}(2, A_k)$.

Proof: In algorithm KVL every element i of $J_{vl}(1, A_k)$ is differ from l_q by at least 2 for every element l_q of $J_1(A_k)$.
So for any $i \in J_{vl}(1, A_k)$ and $l_q \in J_1(A_k), |i - l_q| \geq 2$.
So, Algorithm KVL correctly computes $J_{vl}(1, A_k), k = 2, 3, \ldots, n$.
Again, according to Algorithm KVL any element $l_\alpha \in J_{vl}(2, A_k)$ differs from $l_\beta \in J_2(A_k)$ by at least 1.
Therefore, $|l_l - p_l| \geq 2$ for all $l_s \in J_{vl}(2, A_k)$ and for all $p_t \in J_1(A_k)$ and also $|l_s - p_t| \geq 1$ for all $l_s \in J_{vl}(2, A_k)$ and for all $p_t \in J_2(A_k)$.

Therefore, $J_{vl}(2, A_k)$ is correctly computed by algorithm KVL.

LEMMA **3.1.** *For every CirG G, $J_{vl}(1, A_k)$ is the non-empty largest set of labels satisfying the condition at distance 1 of L21L, where $m \leq s$ for all $m \in J_{vl}(1, A_k), s = \max\{J(A_k)\} + 2$ for any $A_k \in A$(set of arcs of G).*

Proof: Obviously, $J_1(A_k) \subseteq J(A_k)$ and since, $s = \max\{J(A_k)\} + 2$, so $|s - l_p| \geq 2$ for all $l_p \in J_1(A_k)$.

So, $s \in J_{vl}(1, A_k)$ and hence, $J_{vl}(1, A_k)$, is non-empty.

Therefore, $s \in J_{vl}(1, A_k)$ and hence $J_{vl}(1, A_k)$ is non-empty.

Now, let A be any set of labels which satisfies the condition of distance 1 of L21LP, where $m \leq s$ for all $m \in A$.

Also, let $\alpha \in A$. Then $|\alpha - l_p| \geq 2$ for all $l_p \in J_1(A_k)$.

Thus, $\alpha \in J_{vl}(1, A_k)$.

So, $\alpha \in A$ implies $\alpha \in J_{vl}(1, A_k)$.

Therefore, $A \subseteq J_{vl}(1, A_k)$.

Since A is arbitrary, so $J_{vl}(1, A_k)$ is a non-empty largest set.

LEMMA **3.2.** *For every CirG G, $J_{vl}(2, A_k)$ is the non-empty largest set of labels satisfying L21L condition, where $m \leq s$ for all $m \in J_{vl}(1, A_k)$, $s = \max\{J(A_k)\} + 2$ for any $A_k \in A$ set of arcs of G.*

Proof: Obviously, $J_1(A_k) \subseteq J(A_k)$ and $J_2(A_k) \subseteq J(A_k)$ and also since $s = \max\{J(A_k)\} + 2$, so $|s - l_p| \geq 2$ for all $l_p \in J_1(A_k), l_p \in J_2(A_k)$.

That is, $|s - l_p| \geq 2$ for all $l_p \in J_1(A_k)$, and
$|s - l_p| \geq 1$ for all $l_p \in J_2(A_k)$.

So, s is the valid $L(2, 1)$-label for the arc A_k. Thus $J_{vl}(2, A_k)$.

This shows that $J_{vl}(2, A_k)$ is non-empty.

Again, let A be any set of labels which satisfies L21L condition and $m \leq s$ for all $m \in A$.

Also, consider, $\alpha \in A$. Then $|\alpha - l_p| \geq 2$ for all $l_p \in J_1(A_k)$ and $|\alpha - l_q| \geq 1$ for all $l_q \in J_2(A_k)$.

Therefore, $\alpha \in J_{vl}(2, A_k)$.

Hence, $A \subset J_{vl}(2, A_k)$.

Since A is arbitrary, $J_{vl}(2, A_k)$ is non-empty largest set of labels satisfying L21L condition, where $m \leq s$ for all $m \in J_{vl}(2, A_k)$, and $s = \max\{J(A_k)\} + 2$ for any $A_k \in A$.

Let's consider R labels necessary for L21L a CirG G. Now, assume we have $r(> 0)$ labels available, where $r < R$. Consequently, some vertices remain unlabeled. Denote the sets of unlabeled and labeled vertices as V_u and V_l respectively, such that $V_l = V - V_u$.

The aims of Algorithm KL21-CirG is to label all vertices of a CirG by using r labels.

Algorithm KL21-CirG

Input: The set of arcs (vertices) of a CirG, $A = \{A_1, A_2, \ldots, A_n\}$ and $J_{vl}(1, A_k), J_{vl}(2, A_k)$ for $k = 2, 3, \cdots, n$, and r, number of maximum available labels.

Output: The set of labeled vertices, V_l.

Step 1: (Initialization)
$\ell(1) = 0$;
$V_l = \{v_1\}$;
$J(A_2) = \{0\}$;
Step 2:
For $k = 2$ to $n - 1$
  $\ell(k) = \min\{J_{vl}(2, A_k)\}$;
  $J(A_{k+1}) = J(A_k) \cup \{\ell(k)\}$;

if $\ell(k) < r$ then
$V_l = V_l \cup \{\ell(k)\}$;
endfor;
Step 3:
$\ell(n) = \min\{J_{vl}(2, A_n)\}$;
if $\ell(n) < r$ then
$V_l = V_l \cup \{\ell(n)\}$.
end KL21-CirG

Theorem 3.2. *The Algorithm KL21-CirG correctly labels a CirG by L21L using r labels, where $r < R$.*

Proof: Let G be a CirG with vertices $A = \{A_1, A_2, \ldots, A_n\}$. We set $\ell(1) = 0$ and $J(A_2) = \{0\}$.

Consider a scenario where some arcs $A_1, A_2, \ldots, A_{k-1}$ are already labeled for $k = 2, 3, \ldots, n$, and the remaining vertices are unlabeled. Our objective is to label the arc A_k by L21L. According to Lemma 3.2, $J_{vl(2,A_k)}$ represents the largest non-empty set of labels satisfying the L21L condition.

To minimize the label usage, we set $\ell(k) = q$, where $q = \min\{J_{vl}(2, v_k)\}$.

If $q \le r$, then q is a valid r-$L(2,1)$-label for the arc A_k. Since A_k is arbitrary, Algorithm KL21-CirG correctly labels any CirG by L21L using $r(< R)$.

To illustrate the algorithm, let us consider a CirG G shown in Figure 1. In this graph $V = \{v_1, v_2, \cdots, v_8\}$ and $A = \{A_1, A_2, \cdots, A_8\}$.

Assume that $r = 5$.

Iteration: Initially, $\ell(1) = 0, J(A_2) = \{0\}, V_l = \{A_1\}$
Iteration 1: For $k = 2$
$J_1(A_2) = \{0\}, J_2(A_2) = \emptyset$
$J_{vl}(1, A_2) = \{2\}, J_{vl}(2, A_2) = \{2\}$
$\therefore \ell(2) = 2, J(A_3) = \{0, 2\}$
Since, $\ell(2) = 2 < 5$, $V_l = \{A_1, A_2\}$.
Iteration 2: For $k = 3$
$J_1(A_3) = \{0, 2\}, J_2(A_3) = \emptyset$
$J_{vl}(1, A_3) = \{4\}, J_{vl}(2, A_3) = \{4\}$
$\therefore \ell(3) = 4, J(A_4) = \{0, 2, 4\}$
Since, $\ell(3) = 4 < 5$, $V_l = \{A_1, A_2, A_3\}$.
Iteration 3: For $k = 4$
$J_1(A_4) = \{4\}, J_2(A_4) = \{0, 2\}$.
$J_{vl}(1, A_4) = \{0, 1, 2\}, J_{vl}(2, A_4) = \{1\}$
$\therefore \ell(4) = 1, J(A_5) = \{0, 1, 2, 4\}$.
Here, $\ell(4) = 1 < 5$, $V_l = \{A_1, A_2, A_3, A_4\}$
Iteration 4: For $k = 5$
$J_1(A_5) = \{1, 4\}, J_2(A_5) = \{0, 2\}$.
$J_{vl}(1, A_5) = \{6\}, J_{vl}(2, A_5) = \{6\}$
$\therefore \ell(5) = 6, J(A_6) = \{0, 1, 2, 4, 6\}$.
In this case, $\ell(5) = 6 > 5$, $V_l = \{A_1, A_2, A_3, A_4\}$.
Iteration 5: For $k = 6$
$J_1(A_6) = \{6\}, J_2(A_6) = \{1, 4\}$
$J_{vl}(1, A_6) = \{0, 1, 2, 3, 4, 8\}, J_{vl}(2, A_6) = \{0, 2, 3, 8\}$
$\therefore \ell(6) = 0, J(A_7) = \{0, 1, 2, 4, 6\}$
Here, $\ell(6) = 0 < 5$, $V_l = \{A_1, A_2, A_3, A_4, A_6\}$.

Iteration 6: For $k = 7$
$J_1(A_7) = \{0\}, J_2(A_7) = \{0, 6\}$
$J_{vl}(1, A_7) = \{2, 3, 4, 5, 6, 7, 8\}, J_{vl}(2, A_7) = \{2, 3, 4, 5, 7, 8\}$.
$\therefore \ell(7) = 2, J(A_8) = \{0, 1, 2, 4, 6\}$
In this case, $\ell(7) = 2 < 5,\ \ V_l = \{A_1, A_2, A_3, A_4, A_6, A_7\}$.
Iteration 7: For $k = 8$
$J_1(A_8) = \{0, 2\}, J_2(A_8) = \{0, 2, 4\}$
$J_{vl}(1, A_8) = \{4, 5, 6, 7, 8\}, J_{vl}(2, A_8) = \{5, 6, 7, 8\}$.
$\therefore \ell(8) = 5,$
Since, $\ell(5) = 5,\ \ V_l = \{A_1, A_2, A_3, A_4, A_6, A_7, A_8\}$.
Hence, $V_u = V - V_l = \{v_5\}$.

Theorem 3.3. *The running time of Algorithm KL21-CirG for a graph with* n *vertices and a maximum degree* Δ *is* $O(n\Delta^2)$.

Proof: According to our proposed algorithm $\ell(k)$, the $L(2,1)$-label of the arc A_k is found if $J_{vl}(2, A_k)$ is available.

The sets $J_{vl}(i, A_k), i = 1, 2$ are computed by Algorithm KVL using $O(\Delta^2)$ time.

Hence, the total time complexity of this algorithm is $O(n\Delta^2)$.

4. An Application

An application of maximum cardinality r-L21LP is considered here. Nowadays television and online sites became popular and powerful medias both for entertainment, broadcasting and receiving various information, viz. news, any kind of circular, product information, etc. Again, during the last few years a huge number of television channels and online news portals are initiated and as a result the general people facing problems to select the television or online channels. Many good channels are now available that are playing very good entertaining programs. On the other hand, a big competition amongst different governmental, nongovernmental,national, international, channels have increased its popularity. The aims of the channels are to catch the maximum number of audiences to broadcast their programs and generate the maximum amount of revenue. The most of the revenue is coming from the advertisement of the different products for the manufacturing companies and other organizations, institutions, etc.

Some channels, whether television or online, broadcast their programs 24 hours a day. Each channel offers a variety of programs scheduled at different times. However, programs aired on different channels at the same time do not overlap. Typically, an individual cannot watch more than one program simultaneously, especially when it comes to entertainment shows. The viewership for each program can be assessed using various methods, with different programs attracting varying numbers of viewers. The quality of a program plays a crucial role in enticing viewers to watch it, ultimately determining its success in attracting more viewers compared to programs airing simultaneously on other channels.

It is obvious that the aim of the advertisers is to send the product information or other information to the maximum number of people by spending a minimum amount of money. To achieve this goal the advertiser will select some popular television programs or online programs. But, it generally happens that the advertisement cost of the popular programs are high compared to non-popular programs. The advertisers follow different rules for broadcasting their information among the maximum number of people. Some advertisers may decide that they will advertise
(i) for the whole day, but they will select only one program at a time.
(ii) for a certain number of programs, say r number of programs. These programs may be

FIGURE 2. Depicting the program slots outlined in Tables 2, 3, and 4 using circular arcs.

FIGURE 3. Representation of the program slots of Tables 2, 3 and 4 as circular arcs.

disjoint or simultaneously telecasted,
(iii) for some programs with a gap of at least one program,
(iv) for a certain number of programs, but they have a fixed amount of money, say maximum \$ M.

Many different such strategies are adopted by the advertisers depending on the availability of the money, need, etc. In this article, it is assumed that the advertisers will advertise their product or information through some channels in r number of program slots such that no two program slots are consecutive, i.e. between two selected program slots there must at least one program slot in between them which is(are) not selected for advertisement.

This problem can be solved from the output obtained by algorithm KL21-CirG.

It's fascinating to note that we can represent all program slots of all channels as a CirG. Each program slot is visualized as a concentric circular arc spanning the 24-hour duration, with the length of each arc representing the duration of the program. To maintain clarity and avoid confusion, each program seamlessly transitions from the end of the preceding program to the start of the succeeding one. For instance, if program A on a channel airs from 14:30 to 15:00, it's immediately followed by the program starting at 15:00, with no gaps in between. This approach ensures a continuous flow of programming without overlaps or gaps.

Now, we can depict all programs of all channels as arcs in a CirG. Each program corresponds to a distinct arc, with intersecting arcs representing programs that share a common time slot. Notably, programs within the same channel do not overlap in time, resulting in non-intersecting arcs. The number of viewers for each program can be represented by the weight assigned to the corresponding arc in the graph.

4.1. **An Illustration.** To demonstrate the scenario, let's consider three television channels: Sony, BBC, and DD1. Sony and BBC broadcast programs continuously throughout the 24-hour duration, while DD1 remains inactive only during the time interval [23:30, 24:00]. The programs telecasted on these channels are tabulated in Tables 2, 3, and 4, respectively. The number of viewers for each program is provided in millions. For simplicity, programs on Sony are denoted as S_1, S_2, etc., those on BBC as B_1, B_2, etc., and those on DD1 as D_1, D_2, etc.

All the program slots for three channels are depicted in Figure 2.

The graph corresponding to the circular arcs is depicted in Figure 3.

Performing the algorithm KL21-CirG for the graph of Figure 2, the labels on the vertices are determined and the vertices and the labels are shown in Table 5.

From the definition of L21L the following fact is obvious:
If the label difference between two vertices p and q is i, $(i = 0, 1, 2)$, then the distance between them is $3 - i$ or more. This concept is used to find the set of r vertices (program slots) among $n(> r)$ vertices. To find such a set of vertices the following algorithm is used.

Time slots	0.00-2.30	2.30-3.30	3.30-4.30	4.30-5.00
Program name	S_1	S_2	S_3	S_4
No. of viewers	0.5	1.1	1.5	1.4
Time slots	5.00-5.30	5.30-6.30	6.30-9.20	9.20-10.20
Program name	S_5	S_6	S_7	S_8
No. of viewers	1.3	2.5	4	1.5
Time slots	10.20-11.00	11.00-11.30	11.30-12.30	12.30-14.00
Program name	S_9	S_{10}	S_{11}	S_{12}
No. of viewers	3.5	5.5	4.5	8.2
Time slots	14.00-15.30	15.30-16.30	16.30-17.00	17.00-18.00
Program name	S_{13}	S_{14}	S_{15}	S_{16}
No. of viewers	3.5	4.4	4.2	5.3
Time slots	18.00-19.30	19.30-21.00	21.00-22.00	22.00-24.00
Program name	S_{17}	S_{18}	S_{19}	S_{20}
No. of viewers	4.5	7.2	8.5	2.50

TABLE 2. Programs for the TV channel Sony

Time slots	0.00-1.00	1.00-1.40	1.40-3.00	3.00-4.00
Program name	B_1	B_2	B_3	B_4
Number of viewers	2	1.5	2	2.1
Time slots	4.00-5.00	5.00-6.00	6.00-7.00	7.00-8.00
Program name	B_5	B_6	B_7	B_8
Number of viewers	2.5	1.6	2.5	2.2
Time slots	8.00-9.00	9.00-10.00	10.00-11.00	11.00-12.00
Program name	B_9	B_{10}	B_{11}	B_{12}
Number of viewers	4.5	4.2	1.5	2.5
Time slots	12.00-13.00	13.00-14.00	14.00-15.00	15.00-15.40
Program name	B_{13}	B_{14}	B_{15}	B_{16}
Number of viewers	9.5	6.3	5	8.5
Time slots	15.40-17.00	17.00-18.00	18.00-19.00	19.00-20.00
Program name	B_{17}	B_{18}	B_{19}	B_{20}
Number of viewers	5	7	3.	4.2
Time slots	20.00-21.00	21.00-22.00	22.00-23.00	23.00-24.00
Program name	B_{21}	B_{22}	B_{23}	B_{24}
Number of viewers	4.2	3.5	2	1.9

TABLE 3. Programs of the TV channel BBC

Let P_j be the set of vertices with the same labels, $j = 1, 2, \ldots, \lambda$, where λ is the total number of groups of vertices with the same labels. All P_j are non-empty and $|P_j| \geq 1$ for all j. One important observation is stated below:

Note 1. *Let $x \in P_i$ and $y \in P_j$ be two vertices. Then $\ell(x) < \ell(y)$ if and only if $i < j$.*

Algorithm **FindrSlots**
Input: Program slots (vertices) and their labels for the graph $G = (V, E)$ and r.
Output: Set (P) of r program slots.
Step 1: Sort the vertices according to their labels.
Step 2: Select largest P_j and let it be P_α, $\alpha \in \{1, 2, 3, \ldots, \lambda\}$.
Step 3: There are three cases.

Time slots	23.30-2.00	2.00-3.00	3.00-3.30	3.30-4.00
Program name	D_1	D_2	D_3	D_4
Number of viewers	1.8	2.1	3.4	3.3
Time slots	4.00-5.00	5.00-5.30	5.30-6.00	6.00-6.30
Program name	D_5	D_6	D_7	D_8
Number of viewers	2.4	2.8	3.1	2.7
Time slots	6.30-7.00	7.00-7.15	7.15-8.15	8.15-8.30
Program name	D_9	D_{10}	D_{11}	D_{12}
Number of viewers	2.3	3.4	2.1	4.5
Time slots	8.30-9.30	9.30-10.00	10.00-11.00	11.00-12.30
Program name	D_{13}	D_{14}	D_{15}	D_{16}
Number of viewers	2.4	1.5	4.2	4.4
Time slots	12.30-13.30	13.30-14.30	14.30-15.00	15.00-16.00
Program name	D_{17}	D_{18}	D_{19}	D_{20}
Number of viewers	3.2	2.2	3.1	7.1
Time slots	16.00-16.15	16.15-16.45	16.45-20.00	20.00-20.30
Program name	D_{21}	D_{22}	D_{23}	D_{24}
Number of viewers	4.5	4.8	4.7	2.2
Time slots	20.30-21.00	21.00-22.00	22.00-22.30	22.30-23.00
Program name	D_{25}	D_{26}	D_{27}	D_{28}
Number of viewers	0.8	1.4	1.8	1.5

TABLE 4. Programs of the TV channel DD1

Vertex	**Label**	**Vertex**	**Label**	**Vertex**	**Label**	**Vertex**	**Label**
S_1	0	S_2	1	S_3	2	S_4	3
S_5	0	S_6	1	S_7	0	S_8	1
S_9	0	S_{10}	0	S_{11}	1	S_{12}	2
S_{13}	0	S_{14}	1	S_{15}	0	S_{16}	10
S_{17}	2	S_{18}	1	S_{19}	0	S_{20}	1
B_1	2	B_2	4	B_3	3	B_4	5
B_5	0	B_6	3	B_7	4	B_8	2
B_9	3	B_{10}	5	B_{11}	3	B_{12}	3
B_{13}	5	B_{14}	4	B_{15}	3	B_{16}	4
B_{17}	3	B_{18}	4	B_{19}	5	B_{20}	6
B_{21}	4	B_{22}	2	B_{23}	3	B_{24}	5
D_1	7	D_2	6	D_3	7	D_4	8
D_5	6	D_6	5	D_7	6	D_8	7
D_9	6	D_{10}	7	D_{11}	8	D_{12}	9
D_{13}	10	D_{14}	7	D_{15}	6	D_{16}	7
D_{17}	8	D_{18}	6	D_{19}	5	D_{20}	7
D_{21}	5	D_{22}	6	D_{23}	8	D_{24}	7
D_{25}	9	D_{26}	5	D_{27}	6	D_{28}	8

TABLE 5. Vertices and their labels

$\mathbf{P}_1$:	S_1	S_7	S_9	S_{16}	S_{18}	S_{20}	B_5	B_{14}	D_3	D_7	D_{20}		
	0	0	0	0	0	0	0	0	0	0	0		
$\mathbf{P}_2$:	S_{10}	B_6	B_{10}	B_{15}	B_{17}	D_{27}							
	1	1	1	1	1	1							
$\mathbf{P}_3$:	S_3	S_{12}	S_{17}	S_{19}	B_1	B_3	B_7	B_{18}	D_2	D_5	D_{12}	D_{24}	D_{28}
	2	2	2	2	2	2	2	2	2	2	2	2	2
$\mathbf{P}_4$:	S_8	B_2	B_{12}	D_{19}	D_{21}								
	3	3	3	3	3								
$\mathbf{P}_5$:	S_2	S_4	S_6	B_9	B_{13}	B_{16}	B_{19}	B_{21}	B_{23}	D_1	D_4	D_9	D_{11}
	4	4	4	4	4	4	4	4	4	4	4	4	4
$\mathbf{P}_5$:	D_{15}	D_{18}	D_{22}	$\mathbf{P}_6$:	D_{14}	D_{16}	D_{26}						
	4	4	4		5	5	5						
$\mathbf{P}_7$:	S_5	S_{11}	S_{13}	S_{15}	B_4	B_8	B_{20}	B_{24}	D_8	D_{13}	D_{25}		
	6	6	6	6	6	6	6	6	6	6	6		
$\mathbf{P}_8$:	S_{14}	B_{11}	B_{22}	D_6	D_{10}	D_{17}	D_{23}						
	8	8	8	8	8	8	8						

TABLE 6. Ordered program slots according to labels

Step 3.1: If $|P_\alpha| = r$, then P_α is the output, i.e. P.
Step 3.2: If $|P_\alpha| > r$, then remove $|P_\alpha| - r$ program slots from P_α whose number of viewers is less than the other sets and this reduced set is the output, P.
Step 3.3: If $|P_\alpha| < r$. // Need $r - |P_\alpha|$ more program slots//
Let $P^t = P_\alpha$.
Select $P_{\alpha-1}$ or $P_{\alpha+1}$ if they exist. At least one such set must exist if $\lambda > 1$.
Let such a set be P''. Select a vertex x from P'' such that the distance between x and the vertices of P_α is more than 2. Add x to P^t.
Select left or right side set of P'' and repeat this process until $|P^t|$ becomes r.
end **FindrSlots**

Theorem 4.1. *The time to find the r program slots among n slots when program slots are at least two distance apart is $O(n\Delta^2)$, where Δ is the degree of the graph.*

Proof: If the graph has n vertices then the labels of the vertices can be ordered in $O(n)$ time, as the labels are non-negative integers. Selection of the largest P_j can also be done within the same time. So, steps 1 and 2 can be computed in $O(n)$ time. Now, in step 3, there are three cases, among them case 3.1 and 3.2 are trivial. If case 3.1 is true then the time for this step is $O(1)$. Step 3.2 can be performed in $O(n)$ time. There are many operations in step 3.3, though this step can be executed in $O(n)$ time.

Lastly, the labels of the vertices, which is the input of the algorithm, can be determined in $O(n\Delta^2)$.

Hence, the overall time to find the r program slots among n slots is $O(n\Delta^2)$. □

An illustration. For the graph shown in Figure 3, the program slots arranged as per ascending order of the labels are shown in Table 6.

The graph of Figure 3 has 72 vertices, but interestingly it is labeled using only eight labels. Hence, there are eight different sets P_j and they are mentioned in Table 6 and $\lambda = 8$. Note that label 7 is not used. So, this label is called hole for this case.

S_2	S_4	S_6	B_9	B_{13}	B_{16}	B_{19}	B_{21}	B_{23}	D_1	D_4	D_9	D_{11}	D_{15}	D_{18}	D_{22}
1.1	1.4	2.5	4.5	9.5	8.5	3	4.2	2	1.8	3.3	2.3	2.1	4.2	2.2	4.8

TABLE 7. The set P_5 and number of viewers

Now, for given different values of r we different results. Mainly there are three different cases.

Case 1: Let $r = 16$.

In this case, there is only one set with 16 program slots and that is P_5, i.e. $\{S_2, S_4, S_6, B_9, B_{13}, B_{16}, B_{19}, B_{21}, B_{23}, D_1, D_4, D_9, D_{11}, D_{15}, D_{18}, D_{22}\}$.

Other different solutions may be obtained by applying Step 3.3 of Algorithm FindrSlots.

Case 2: Let $r = 13$.

In this case, there is one readily available solution which is P_3, i.e. $\{S_3, S_{12}, S_{17}, S_{19}, B_1, B_3, B_7, B_{18}, D_2, D_5, D_{12}, D_{24}, D_{28}\}$.

Other solutions can be determined from the set P_5 by removing three program slots. We can remove any three program slots, but to spread out the advertisement to more people, we remove such program slots whose number of viewers are less (see Table 7).

From Table 7 it follows that the programs S_2, S_4 and D_1 have the least number of viewers, viz. 1.1, 1.4 and 1.8. So, we can delete these programs and obtained another solution $\{S_6, B_9, B_{13}, B_{16}, B_{19}, B_{21}, B_{23}, D_4, D_9, D_{11}, D_{15}, D_{18}, D_{22}\}$.

Again, another solution may be obtained by applying the Step 3.3 of Algorithm FindrSlots.

5. Discussion

The r-L21LP is investigated for CirGs. An algorithm whose time complexity $O(n\Delta^2)$ is designed for finding the solution to this problem. Also, an application is presented to choose the r program slots among $n(> r)$ program slots from telecasting channels. A polynomial time algorithm is presented to solve this problem. The application is illustrated by real data collected from three television channels. This method can be employed to identify r (a specified integer) disjoint program slots that attract the highest number of viewers. Also, cost of advertisement may be incorporated with the program slots. Then bi-objective problem, i.e. maximized number of viewers and minimized advertising cost, can be modelled and solved by graph theoretic approach. The r-L(2,1)-labeling problem on circular-arc graphs faces scalability issues, as finding an optimal labeling becomes computationally intensive with increasing graph size, making it difficult to handle very large graphs efficiently. Additionally, the restriction to labels within $0, 1, 2, \ldots, r$ can limit the flexibility of the labeling scheme, often resulting in suboptimal coverage in complex or densely connected graphs. In future we will try to solve this type of problems using another classes of intersection graph.

Acknowledgement. We gratefully acknowledge the financial support provided to the first author by DHESTBT (Government of West Bengal, India), under Ref. No. 245(Sanc.)/ST/P/S&T/16G-20/2017.

References

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

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

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

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

[5] Bera, B., Amanathulla, S., Pal, M., (2022), $L(2,1,1)$-labeling of interval graphs, International Journal of Mathematics for Industry, 14(1), pp. 22500031-22500038.

[6] Barman, S. C., 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), pp. 743-755.

[7] Bera, B., Amanathulla, S., Mahato, S. K., 2023. A Comprehensive Study of Picture Fuzzy Planar Graphs with Real-World Applications, Journal of Uncertain systems, doi.org/10.1142/S1752890932500095.

[8] Bera, D., Pal, T. K., Pal, M., (1999), An optimal parallel algorithm to compute all cutvertices and blocks on permutation graphs, International Journal of Computer Mathematics, 72(4), pp. 449-462.

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

[10] Adhikari, B. D., Banerjee, A., Amanathulla, S., Mondal, S., (2023), Certain Operations on Interval-Valued Picture Fuzzy Graphs with Application, International Journal of Mathematics for Industry, doi.org/10.1142/S2661335223500089.

[11] 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.

[12] Borzooei, R. A., Rashmanlou, H., Samanta, S., Pal, M., (2016), New concepts of vague competition graphs, Journal of Intelligent & Fuzzy Systems, 31 (1), pp. 69-75.

[13] 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), pp. 27-34.

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

[15] Chang, G. J., Kuo, D., (1996), The $L(2,1)$-labelling on graphs, SIAM J. Discrete Math., 9, pp. 309-316.

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

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

[18] Ghosh, P.K., Pal, M., (2007), An efficient algorithm to solve connectivity problem on trapezoid graphs, Journal of Applied Mathematics and Computing, 24, pp. 141-154.

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

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

[21] Hale, W. K., (1980), Frequency Assignment: Theory and Applications, Proc. IEEE, 68, pp. 1497-1514.

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

[23] Khatun, J., Amanathulla, S, Peng, S. L., (2023), Minimization of Span in L(3, 1)-Labeling for a Particular Type of Intersection Graphs, in book Fuzzy Optimization, Decision-making and Operations Research: Theory and Applications, Springer International Publishing, pp. 207-217.

[24] Khatun, J., Amanathulla, S., Pal, M. (2023). Picture fuzzy cubic graphs and their applications. Journal of Intelligent & Fuzzy Systems, pp. 1–18.

[25] Khatun, J., Amanathulla, S., (2023). An Application of Neutrosophic Graph in Decision-Making Problem for Alliances of Companies, in Book Fuzzy Optimization, Decision -making and Operations Research: Theory and Applications, Springer International Publishing, pp. 241-255.

[26] Patra, N., Amanathulla, S., Pal, M., Mondal, S., (2023), A restricted L(2, 1)-labelling problem on interval graphs, TWMS Journal Of Applied And Engineering Mathematics, 13(2), pp. 635-648.

[27] Islam, S. R., Pal, M., (2023), Hyper-connectivity index for fuzzy graph with application, Işık University Press.

[28] Mahapatra, R., Samanta, S., Pal, M., Xin, Q., (2019), RSM index: A new way of link prediction in social networks, Journal of Intelligent and Fuzzy Systems, 37(2), pp. 2137–2151.

[29] Mahapatra, R., Samanta, S., Pal, M., (2020), Applications of edge colouring of fuzzy graphs, Informatica (Netherlands), 31(2), pp. 313–330.

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

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

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

[33] 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.

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

[35] Paul, S., Pal, M., Pal, A., (2014), A linear time algorithm to compute square of interval graphs and their colouring, AKCE International Journal of Graphs and Combinatorics, 13(1), pp. 54-64.

[36] Raut, S., Pal, M., Ghorai, G., (2018), Fuzzy permutation graph and its complements, Journal of Intelligent & Fuzzy Systems 35 (2), pp. 2199-2213.

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

[38] Rana, A., Pal, A., Pal, M., (2012), The conditional covering problem on unweighted interval graphs with nonuniform coverage radius, Mathematics in Computer Science, 6 (1), pp. 33-41.

[39] Rashmanlou, H., Muhiuddin, G., Amanathulla, Sk., Mofidnakhaei, F., Pal, M., (2021), A study on cubic graphs with novel application, Journal of Intelligent and Fuzzy Systems, 40(1), pp. 89–101.

[40] Mondal, U., Mahapatra, T., Xin, Q., Pal, M., (2023), Solution of road network problem with the help of m-polar fuzzy graph using isometric and antipodal concept, Scientific Reports, 13(1), 6452.

[41] Mondal, U., Mahapatra, T., Xin, Q., Pal, M., (2023), Generalized m-polar fuzzy planar graph and its application, IEEE Access, 11, pp. 138399-138413.

[42] Mondal, U., Pal, M., Das, K. (2024), Interval-valued intuitionistic fuzzy soft graphs with application, TWMS J. App. and Eng. Math., 14(1), pp. 1-18.

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

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

[45] Saha, A., Pal, M., Pal, T. K., (2005), An optimal parallel algorithm to find all-pairs shortest paths on circular-arc graphs, J. Applied Mathematics and Computing, 17(1-2), pp. 1-23.

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

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

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

[49] 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.

[50] 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.

[51] 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(1), pp. 467-479.

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

[53] 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, 14, pp. 205-215.

[54] 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.

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

[56] 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), pp. 515-527.

[57] Amanathulla, S., Pal, M., Sahoo, S., (2018), $L(3,1,1)$-labeling numbers of squares of paths, complete graphs and complete bipartite graphs, Journal of Intelligent and Fuzzy Systems, DOI:10.3233/JIFS-172195.

[58] Amanathulla, S., Pal, M., (2018), $L(h,k)$-labelling problems on intersection graphs, In Handbook of Research on Advanced Applications of Graph Theory in Modern Society, IGI Global, pp. 437-468.

[59] Amanathulla, S., Pal, M., (2021), $L(3,2,1)$-labeling problems on Trapezoid graphs, Discrete Mathematics, Algorithms and Applications, 13(05), doi.org/10.1142/S1793830921500683.
[60] 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, pp. 1-9.
[61] Yeh, R. K., (2006), A survey on labelling graphs with a condition at distance two, Discrete Mathematics, 306, pp. 1217-131.
[62] Yea, R. K., (1990), Labeling graphs with condition at distance two, Ph.D Thesis, University of South Carolina, Columbia.
[63] Wan, P. J., (1997), Near-Optimal Conflit-free channel set assignment for an optical cluster-based hypercube network, J. Combin. Optim., 1, pp. 179-186.

Dr. Nupur Patra is an Assistant Professor in the department of Mathematics, Raja N.L. Khan Women's College, Midnapore, India. Her research interests include fuzzy graph theory, graph theory.

Dr. Sk Amanathulla is an Assistant Professor in the department of Mathematics, Raghunathpur College, Purulia, India. His research interests include fuzzy graph theory, graph theory, graph coloring.

Dr. Sukumar Mondal is an Associate Professor in the department of Mathematics, Raja N.L. Khan Women's College, Midnapore, India. His research interests include computational graph theory, fuzzy graph theory.

Dr. Madhumangal Pal for the photography and short autobiography, see TWMS J. App. and Eng. Math. V.9, N.3.