



Efficient Algorithm For $L(3, 2, 1)$ -Labeling of Cartesian Product Between Some Graphs

Sumonta Ghosh^{1,*}, Prakhar Pogde^{1,†}, Narayan C. Debnath^{2,‡}, and Anita Pal^{1,§}

¹ National Institute of Technology Durgapur, West Bengal, India.
mesumonta@gmail.com

² National Institute of Technology Durgapur, West Bengal, India.
prakhark5nov@gmail.com

³ School of Computing and Information Technology, Eastern International University, Vietnam.
narayan.debnath@eiu.edn.vn

⁴ National Institute of Technology Durgapur, West Bengal, India.
anita.buie@gmail.com

Abstract

$L(h, k)$ Labeling in graph came into existence as a solution to frequency assignment problem. To reduce interference a frequency in the form of non negative integers is assigned to each radio or TV transmitters located at various places. After $L(h, k)$ labeling, $L(h, k, j)$ labeling is introduced to reduce noise in the communication network. We investigated the graph obtained by Cartesian Product between Complete Bipartite Graph with Path and Cycle, i. e., $K_{m,n} \times P_r$ and $K_{m,n} \times C_r$ by applying $L(3, 2, 1)$ Labeling. The $L(3, 2, 1)$ Labeling of a graph G is the difference between the highest and the lowest labels used in $L(3, 2, 1)$ and is denoted by $\lambda_{3,2,1}(G)$ In this paper we have designed three suitable algorithms to label the graphs $K_{m,n} \times P_r$ and $K_{m,n} \times C_r$. We have also analyzed the time complexity of each algorithm with illustration.

1 Introduction

In real world, graph labeling actually deals with the frequency assignment problem which can be solved by vertex coloring problem. Requirement of noise free solution station demands efficient frequency assignment techniques. Among various frequency assignment techniques $L(h, k)$ labeling covers some important and realistic domain where some work is done by many researchers. To avoid interference any two "close" transmitters must receive different channels and any two "very close" transmitters must receive channels that are at least two channels apart. In the language of Graph Theory, the transmitters are represented by the vertexes of the graph; two vertexes are "very close" if they are adjacent and

*Primary researcher designed and implemented the algorithm and studied various aspects of communication networks and implementation of graph theory in communication

†Did numerous tests and provided a lot of suggestions

‡Contributed on developing efficient algorithm

§Guide and motivator

"close" if they are at distance two in the graph. This problem is considered as $L(2,1)$ labeling on a simple graph. Graph $G = (V, E)$ has various bound of $\lambda_{2,1}(G)$ is known in terms of Δ , $\omega(G)$ and $\chi(G)$. Maximum degree denoted by Δ , where $\omega(G)$ and $\chi(G)$ denote the size of the maximum clique and chromatic number of the graph G respectively. Griggs and Yeh [2] gives the explanation that a graph required $\Delta^2 - \Delta$ span, in 1992 they also prove that $\lambda_{2,1}(G) \leq \Delta^2 + 2\Delta$. In 2008 Gonclaves [4] improved the bound to $\lambda_{2,1}(G) \leq \Delta^2 + 2\Delta - 2$ and later the bound is improved to $\lambda_{2,1}(G) \leq \Delta^2 + 2\Delta - 3$ for 3-connected graph. Then Chang and Kuo [5] improved the bound to $\lambda_{2,1}(G) \leq \Delta^2 + \Delta$. The conjecture of Griggs and Yeh[2] stable it for the graph of diameter 2. Later Chang and Kuo improved the bound to $\lambda_{2,1}(G) \leq \Delta^2$.

Conjecture 1.1. : For any graph $G = (V, E)$ with maximum degree $\Delta \geq 2$, $\lambda_{2,1}(G) \leq \Delta^2$.

The above conjecture of Griggs and Yeh [2] worked for the set of graphs like path[2], wheel[2], cycle[2], trees[2, 5], co-graphs [5], interval graphs [5], chordal graphs, permutation graph [6], circular arc graph, Cartesian product of complete bipartite graph, path and cycle [14, 15, 16] etc. The bound $\lambda_{2,1}(G)$ can be computed systematically for some graphs like cycle, path, tree[2, 5].

Distance between deployed stations are not fixed geographically. In some cases station 1 and 3 (at distance two apart on the graph) can be at an intermediate distance on earth e.g. when they are at right angles, therefore in such cases $L(h, k)$ labeling fails to assign optimal frequency difference without violating the span. Therefore, interference between two adjacent and two distance apart station arises and may vary accordingly. Our main aim is to achieve a noise free network prior to reduced frequency. To resolve this problem we switch to $L(3, 2, 1)$ labeling. For any graph $G = (V, E)$ Clipperton et al. [7] studied that $\lambda_{3,2,1} \leq \Delta^3 + \Delta^2 + 3\Delta$, later on Chai et al. [8] tune up the upper bound of Clipperton et al. [7] and established that $\lambda_{3,2,1} \leq \Delta^3 + 2\Delta$ for any graph $G = (V, E)$. Chai et al. [8] have also studied that $\lambda_{3,2,1} = 2n + 5$ for complete n -array tree of height ≥ 3 and $2\Delta + 1 \leq \lambda_{3,2,1} \leq 2\Delta + 3$. $\lambda_{3,2,1}$ also been investigated on planar graph by Lui and Shao [9] and they established that $\lambda_{3,2,1} \leq 15(\Delta^2 - \Delta + 1)$. Jean [10] investigate slightly different form of $L(3, 2, 1)$, $L(d, 2, 1)$ on complete graph and complete bipartite graph and established that $\lambda_{d,2,1}(K_n) = d(n-1) + 1$ and $\lambda_{d,2,1}(K_{m,n}) = d + 2(m+n) - 3$. $\lambda_{3,2,1}$ also has been implemented on Cartesian product between complete graph K_3 and cycle C_n by Kim et al. [11] and they established $\lambda_{3,2,1}(K_3 \times C_n) = 15$ for $n \geq 28$ and $n \equiv 0 \pmod{5}$. Clipperton [12] apply $L(4, 3, 2, 1)$ on some simple graph and shown that $\lambda_{4,3,2,1} \leq \Delta^3 + 2\Delta^2 + 6\Delta$. $\lambda_{4,3,2,1}$ and $\lambda_{3,2,1}$ of circular-arc graph and permutation graph has investigated by Amanathulla and Pal [17, 18], they have established $\lambda_{4,3,2,1} = 16\Delta - 12$ and $\lambda_{3,2,1} = 9\Delta - 6$ for any circular-arc graph and $\lambda_{4,3,2,1} = 13\Delta - 18$ and $\lambda_{3,2,1} = 11\Delta - 8$ for any permutation graph.

Here we consider the graph obtained by Cartesian product between complete bipartite graph with path and cycle i.e $G = K_{m,n} \times P_r$ and $G = K_{m,n} \times C_r$. We apply $L(3, 2, 1)$ labeling on the above said graphs. In real life scenarios we consider the Cartesian product of most of the networks are complex and hybrid. We investigate $\lambda_{3,2,1}$ for each graph and established a relation in term of m, n, r . To accomplish the above said task we designed four suitable and efficient algorithms, that is less tedious (in comparison to the traditional trial and error method) and can simplify the assignment of frequencies to the vertexes and we also describe the time complexity of each algorithm.

The remaining part of the paper is organized as follows. Section 2 presents some preliminaries, definition and lemmas, Section 3 describes algorithms, analysis of each algorithm and illustration followed by conclusion.

2 Preliminaries

Definition 2.1. A cycle of a graph $G = (V, E)$ is denoted by C_r , where $V = \{v_0, v_1, \dots, v_r\}$ be the set of vertexes and $E = \{e_0, e_1, \dots, e_r\}$ be the set of edges will form a cycle if every vertex say v_i is adjacent to

exactly two vertexes.

Definition 2.2. Cartesian product of two graphs G and H , denoted by $G \times H$, where $G = (V, E)$ and $H = (V', E')$ be the two graphs, and is defined by taking Cartesian product between two set of vertexes $V(G) \times V'(H)$, where (u, u') and (v, v') are the order pair of the Cartesian product will be directly connected in $G \times H$ if and only if either

1. $u = v$ and u' is directly connected with v' in H , or
2. $u' = v'$ and u is directly connected with v in G .

Lemma 2.1. Let Δ be the degree of the graph $K_{m,n} \times P_r$, then

$$\Delta = \begin{cases} m+1 & \text{for } m > n \text{ and } r = 2 \\ m+2 & \text{for } m > n \text{ and } r > 2 \\ m+1 & \text{for } m = n \text{ and } r = 2 \\ m+2 & \text{for } m = n \text{ and } r > 2 \end{cases} \quad (1)$$

Lemma 2.2. Let Δ be the degree of the graph $K_{m,n} \times C_r$, then

$$\Delta = \begin{cases} m+2 & \text{for } m > n \\ n+2 & \text{for } n > m \end{cases} \quad (2)$$

Lemma 2.3. If H is a sub graph of G , then $\lambda_{3,2,1}(H) \leq \lambda_{3,2,1}(G)$.

Theorem 2.1. For any complete bipartite graph $\lambda_{3,2,1}(K_{m,n}) = 2(m+n)$.

Theorem 2.2. [7] For any cycle, C_n with $n > 3$,

$$\lambda_{3,2,1}(C_n) = \begin{cases} 7, & \text{if } n = 3; \\ 8, & \text{if } n \text{ is even;} \\ 9, & \text{if } n \text{ is odd and } n \neq 3, 7; \\ 10, & \text{if } n = 7. \end{cases} \quad (3)$$

Lemma 2.4. [7] For cycle with length 4, $\lambda_{3,2,1}(C_4) = 8$.

Lemma 2.5. [7] For cycle with length 5, $\lambda_{3,2,1}(C_5) = 9$.

Lemma 2.6. [7] For cycle with length 6, $\lambda_{3,2,1}(C_6) = 8$.

Lemma 2.7. [7] For cycle with length 7, $\lambda_{3,2,1}(C_7) = 10$.

Theorem 2.3. [8] If G is a graph with maximum degree Δ , $\lambda_{3,2,1}(G) = \Delta^3 + 2\Delta$.

3 $L(3,2,1)$ –Labeling

In introduction section we discussed about various types of labeling of trivial graphs, intersection graphs and Cartesian product of some graphs with their bounds $\lambda_{3,2,1}(G)$ in the form of Δ and number of vertexes. In this section we discussed about the $L(3,2,1)$ labeling of Cartesian product between complete bipartite graph with path and Cartesian product between complete bipartite graph with cycle. We have developed two suitable algorithms for $L(3,2,1)$ labeling of the above said graphs. For each algorithm, we analyzed the algorithm along with correctness of proof followed by illustration.

3.1 Algorithm L321P

To label the path P_r of length r with the vertexes v_1, v_2, \dots, v_r by $L(3,2,1)$ labeling we consider one array $V[r]$ to store the labeling value. For $r = 1$, $f(v_1) = 1$ and $r = 2$, $f(v_1) = 1$ and $f(v_2) = 4$. The technique of $L(3,2,1)$ labeling of path P_r for $r > 2$ are given below.

Algorithm 1 $L(3,2,1)$ -Labeling of Path (L321P)

```

1: Input:  $G = P_r$ .
2: Output: Labelled graph  $G = P_r$ .
3: Initialization: Array  $V[r]$ .
4: loop  $i = 1$  to  $r$ 
5:   if  $i - 1$  is divisible by 8 then
6:      $V[i] = 3$ .
7:   end if
8:   if  $i - 2$  is divisible by 8 then
9:      $V[i] = 6$ .
10:  end if
11:  if  $i - 3$  is divisible by 8 then
12:     $V[i] = 1$ .
13:  end if
14:  if  $i - 4$  is divisible by 8 then
15:     $V[i] = 4$ .
16:  end if
17:  if  $i - 5$  is divisible by 8 then
18:     $V[i] = 7$ .
19:  end if
20:  if  $i - 6$  is divisible by 8 then
21:     $V[i] = 2$ .
22:  end if
23:  if  $i - 7$  is divisible by 8 then
24:     $V[i] = 5$ .
25:  end if
26:  if  $i - 8$  is divisible by 8 then
27:     $V[i] = 8$ .
28:  end if
29: end loop

```

3.2 $L(3,2,1)$ -Labeling of $G = K_{m,n} \times P_r$

To develop the algorithm 2 of $L(3,2,1)$ labeling of $G = K_{m,n} \times P_r$ we consider an array $P_{arr}[k]$ to store the $L(3,2,1)$ labeling of path P_r formed by the first vertex of set X for each copy of $K_{m,n}$. We introduced arrays $X_k[i]$ and $Y_k[j]$ for each copy of $K_{m,n}$ for $i = 1, 2, 3, \dots, m$, $j = 1, 2, 3, \dots, n$ and $k = 1, 2, 3, \dots, r$. According to the figure ?? we consider the path formed by the array elements $X_k[0]$, $k = 1, 2, 3, \dots, r$, which is the first vertex of set X for each copy of $K_{m,n}$. We also introduced the variables $maxX$ and $maxY$ initializes with the value $(2m - 1)$ and $2(m + n)$ respectively and initializes $X_{k-1}[1] = 0$ and $Y_{k-1}[1] = (maxX + 2)$. We call the algorithm L321P before start labeling the graph $G = K_{m,n} \times P_r$.

Correctness of L321CP is given below

Algorithm 2 $L(3, 2, 1)$ labeling of the graph $G = K_{m,n} \times P_r$ ($L321CP$)

```

1: Input: Graph  $G = K_{m,n} \times P_r$ .
2: Output:  $L(3, 2, 1)$ -Labelled graph  $G = K_{m,n} \times P_r$ .
3: Initialization:  $maxX \leftarrow (2m - 1)$ ,  $maxY \leftarrow 2(m + n)$ ,  $X_{k-1}[1] \leftarrow 0$ ,  $Y_{k-1}[1] \leftarrow (maxX + 2)$ .
4:  $P_{arr}[k]$  = Call algorithm  $L321P$ .
5: loop  $k = 1$  to  $r$ 
6:    $p = P_{arr}[k]$ .
7:   loop  $i = 1$  to  $m$ 
8:     if  $p \geq maxX$  then
9:       if  $p$  is divisible by 2 then
10:         $p = 0$ .
11:       else  $p = 1$ .
12:       end if
13:     end if
14:      $X_k[i] = p$ .
15:      $p = p + 2$ .
16:   end loop
17:   if  $p$  is divisible by 2 then
18:      $maxX = 2(m - 1)$ .
19:   else  $maxX = 2m - 1$ .
20:   end if
21:    $q = Y_{k-1}[1] + (X_k[1] - X_{k-1}[1])$ .
22:   loop  $j = 1$  to  $n$ 
23:      $Y_k[j] = q$ .
24:      $q = q + 2$ .
25:     if  $q > maxY$  then
26:        $q = maxX + 3$ .
27:     end if
28:   end loop
29: end loop

```

Theorem 3.1. Algorithm $L321CP$ exactly label the graph $G = (K_{m,n} \times P_r)$.

Proof. From theorem 2.1 it is clear that $\lambda_{3,2,1}(K_{m,n}) = 2(m+n)$. We have considered the graph obtained by the Cartesian product between complete bipartite graph and path and we get the graph $G = (K_{m,n} \times P_r)$, it is obvious that the graph $G = K_{m,n} \times P_r$ have r copies of $K_{m,n}$. We have taken two arrays X and Y for each copy of $K_{m,n}$ to store the label. We have considered the array P_{array} to label the path formed by the each first vertex of set X in the graph $G = (K_{m,n} \times P_r)$. We have also taken variables $maxX$ and $maxY$ to store the maximum label for the vertex set X and vertex set Y respectively and p to store the value from the array P_{array} . In the array $P_{array}[r]$, first element is the label for the first vertex of vertex set X for first copy of $K_{m,n}$ i.e $X_1[1] = P_{array}[1]$, second element is the label for the first vertex of vertex set X for second copy of $K_{m,n}$ i.e $X_2[1] = P_{array}[2]$, similarly r^{th} element is the label for the first vertex of vertex set X for r^{th} copy of $K_{m,n}$ i.e $X_r[1] = P_{array}[r]$. Now we fetch the first element of the array $P_{array}[r]$ and start labeling the first copy of $K_{m,n}$ according to algorithm 2 by increasing the label by 2 as all the vertexes in the set X are at distance two. Then fetch the second element from the array P_{array} and start labeling the second copy of $K_{m,n}$. When label of vertex set $X_k[i]$ reaches $maxX$ and p is even then we assign the label $p = 0$ and for odd value of p we assign the label $p = 1$. Similarly when vertex set

Y reaches $\max Y$ then we assign the label $q = \max X + 3$. Similarly we can repeat the remaining steps. We can get the labeling for the graph $G = (K_{m,n} \times P_r) = 2(m+n)$ which is equal to $L(3,2,1)$ labeling of $K_{m,n}$ i.e $\lambda_{3,2,1}(K_{m,n}) = \lambda_{3,2,1}(K_{m,n} \times P_r) = 2(m+n)$. \square

3.2.1 Time Complexity Analysis of Algorithm L321CP:

According to the algorithm L321CP, first loop will work for the path length r then for each encounter next two serial inner loop will run for m times and n times to label each complete bipartite graph $K_{m,n}$. So, to exact label the whole graph that obtained by doing Cartesian product between complete bipartite graph and path is either $O(mr)$ or $O(nr)$ depending on the value of m and n .

3.2.2 $L(3,2,1)$ labeling of Cartesian Product Between Complete Bipartite Graph and Path Satisfies Theorem 2.3

Theorem 3.2. *Algorithm L321CP satisfies the theorem 2.3.*

Case:1 For $m > n$ and $r = 2$

Proof. For the graph $G = (K_{m,n} \times P_r)$ it is shown in the lemma 2.1 that maximum degree is $\Delta = m + 1$. From the theorem 3.1 we got that $\lambda_{3,2,1}(K_{m,n} \times P_r) = 2(m+n)$. As we know

$$\begin{aligned} \Delta^3 + 2\Delta = (m+1)^3 + 2(m+1) &> (m+1)^2 \\ &> m^2 + 2m \quad \text{as } m, n > 0 \\ &\geq 2n + 2m \quad \text{as } m > n \text{ so } m^2 \geq 2n \\ &\geq 2(m+n). \end{aligned}$$

Hence the proof. \square

Case:2 For $m > n$ and $r > 2$

Proof. For the graph $G = (K_{m,n} \times P_r)$ it is shown in the lemma 2.1 that maximum degree is $\Delta = m + 2$. From the theorem 3.1 we got that $\lambda_{3,2,1}(K_{m,n} \times P_r) = 2(m+n)$. As we know

$$\begin{aligned} \Delta^3 + 2\Delta = (m+2)^3 + 2(m+2) &> (m+2)^2 \\ &> m^2 + 2m \quad \text{as } m, n > 0 \\ &\geq 2n + 2m \quad \text{as } m > n \text{ so } m^2 \geq 2n \\ &\geq 2(m+n). \end{aligned}$$

Hence the proof. \square

Case:3 For $m = n$ and $r = 2$

Proof. For the graph $G = (K_{m,n} \times P_r)$ it is shown in the lemma 2.1 that maximum degree is $\Delta = m + 1$. From the theorem 3.1 we got that $\lambda_{3,2,1}(K_{m,n} \times P_r) = 2(m+n)$. As we know

$$\begin{aligned} \Delta^3 + 2\Delta = (m+1)^3 + 2(m+1) &> (m+1)^2 \\ &> m^2 + 2m \quad \text{as } m, n > 0 \\ &\geq 2n + 2m \quad \text{as } m = n \text{ so } m^2 \geq 2n \\ &\geq 2(m+n). \end{aligned}$$

Hence the proof. \square

Case:4 For $m = n$ and $r > 2$

Proof. For the graph $G = (K_{m,n} \times P_r)$ it is shown in the lemma 2.1 that maximum degree is $\Delta = m + 2$. From the theorem 3.1 we got that $\lambda_{3,2,1}(K_{m,n} \times P_r) = 2(m + n)$. As we know

$$\begin{aligned} \Delta^3 + 2\Delta = (m+2)^3 + 2(m+2) &> (m+2)^2 \\ &> m^2 + 2m \quad \text{as } m, n > 0 \\ &\geq 2n + 2m \quad \text{as } m = n \text{ so } m^2 \geq 2n \\ &\geq 2(m+n). \end{aligned}$$

Hence the proof. □

Theorem 3.3. For any sub-graph S of the graph $G = (K_{m,n} \times P_r)$, $\lambda_{3,2,1}(S) \leq 2(m + n)$.

Proof. We got the span of the graph $G = (K_{m,n} \times P_r)$ is $\lambda_{3,2,1}(K_{m,n} \times P_r) = 2(m + n)$. As S be the sub-graph of G , so according to lemma 2.3 we can conclude that $\lambda_{3,2,1}(S) \leq 2(m + n)$. Hence the proof. □

3.3 $L(3,2,1)$ –Labeling of $G = K_{m,n} \times C_r$

To develop the algorithm 2 of $L(3,2,1)$ labeling of $G = K_{m,n} \times C_r$ we consider an array $C_{arr}[k]$ to store the $L(3,2,1)$ labeling of cycle C_r form by the first vertex set X for each copy of $K_{m,n}$. We introduced arrays $X_k[i]$ and $Y_k[j]$ for each copy of $K_{m,n}$ for $i = 1, 2, 3, \dots, m$, $j = 1, 2, 3, \dots, n$ and $k = 1, 2, 3, \dots, r$. According to the figure ?? we consider the cycle formed by the array elements $X_k[0]$, $k = 1, 2, 3, \dots, r$, which is the first vertex set X for each copy of $K_{m,n}$. We also introduced the variables $maxX$ and $maxY$ initializes with the value $(2m - 1)$ and $2(m + n)$ respectively and initializes $X_{k-1}[1] = 0$ and $Y_{k-1}[1] = (maxX + 2)$. From the theorem 2.2 we insert label in the array $C_{array}[k]$ before start labeling the graph $G = K_{m,n} \times C_r$.

Correctness of L321CC is given below

Theorem 3.4. Algorithm L321CC exactly label the graph $G = (K_{m,n} \times C_r)$.

Proof. From theorem 2.1 it is clear that $\lambda_{3,2,1}(K_{m,n}) = 2(m + n)$. We have considered the graph obtained by the Cartesian product between complete bipartite graph and cycle and we get the graph $G = (K_{m,n} \times C_r)$, it is obvious that the graph $G = K_{m,n} \times C_r$ have r copies of $K_{m,n}$. We have taken two arrays X and Y for each copy of $K_{m,n}$ to store the label. We have considered the array C_{array} to label the path formed by the first vertex of set X in the graph $G = (K_{m,n} \times C_r)$. We also have taken variables $maxX$ and $maxY$ to store the maximum label for the vertex set X and vertex set Y respectively and p to store the value from the array C_{array} . In the array $C_{array}[r]$, first element is the label for the first vertex of vertex set X for first copy of $K_{m,n}$ i.e $X_1[1] = C_{array}[1]$, second element is the label for the first vertex of vertex set X for second copy of $K_{m,n}$ i.e $X_2[1] = C_{array}[2]$, similarly r^{th} element is the label for the first vertex of vertex set X for r^{th} copy of $K_{m,n}$ i.e $X_r[1] = C_{array}[r]$. Now we fetch the first element of the array $C_{array}[r]$ and start labeling the first copy of $K_{m,n}$ according to algorithm 3 by increasing the label by 2 as all the vertexes in the set X are at distance two. Then fetch the second element from the array C_{array} and start labeling the second copy of $K_{m,n}$ similarly. When label for vertex set X reached $maxX$ and p is even, then we assign the label $p = 0$ and for odd value of p we assign the label $p = 1$. Similarly when vertex set Y reached $maxY$ then we assign the label $q = maxX + 3$. Similarly we can follow remaining steps. We can obtain the labeling for the graph $G = (K_{m,n} \times C_r) = 2(m + n)$ which is equal to $L(3,2,1)$ labeling of $K_{m,n}$ i.e $\lambda_{3,2,1}(K_{m,n}) = \lambda_{3,2,1}(K_{m,n} \times C_r) = 2(m + n)$. □

Algorithm 3 $L(3,2,1)$ labeling of the graph $G = K_{m,n} \times C_r$ (L321CC)

```

1: Input: Graph  $G = K_{m,n} \times C_r$ .
2: Output:  $L(3,2,1)$ -Labelled graph  $G = K_{m,n} \times C_r$ .
3: Initialization:  $maxX \leftarrow (2m - 1)$ ,  $maxY \leftarrow 2(m + n)$ ,  $X_{k-1}[1] \leftarrow 0$ ,  $Y_{k-1}[1] \leftarrow (maxX + 2)$ .
4:  $C_{arr}[k] = L(3,2,1)$  labeling of cycle by using the theorem 2.2.
5: loop  $k = 1$  to  $r$ 
6:    $p = C_{arr}[k]$ .
7:   loop  $i = 1$  to  $m$ 
8:     if  $p \geq maxX$  then
9:       if  $p$  is divisible by 2 then
10:         $p = 0$ .
11:       else  $p = 1$ .
12:       end if
13:     end if
14:      $X_k[i] = p$ .
15:      $p = p + 2$ .
16:   end loop
17:   if  $p$  is divisible by 2 then
18:      $maxX = 2(m - 1)$ .
19:   else  $maxX = 2m - 1$ .
20:   end if
21:    $q = Y_{k-1}[1] + (X_k[1] - X_{k-1}[1])$ .
22:   loop  $j = 1$  to  $n$ 
23:      $Y_k[j] = q$ .
24:      $q = q + 2$ .
25:     if  $q > maxY$  then
26:        $q = maxX + 3$ .
27:     end if
28:   end loop
29: end loop

```

3.3.1 Time Complexity of Algorithm L321CC:

According to the algorithm L321CC, first loop will work for the path length r then for each encounter next two serial inner loop will run for m times and n times to label each complete bipartite graph $K_{m,n}$. So, to exact label the whole graph that obtained by doing Cartesian product between complete bipartite graph and path is either $O(mr)$ or $O(nr)$ depending on the value of m and n .

3.3.2 $L(3,2,1)$ labeling of Cartesian Product Between Complete Bipartite Graph and Cycle Satisfies Theorem 2.3

Theorem 3.5. Algorithm L321CC satisfy the theorem 2.3.

Case:1 For $m > n$ and for any length of cycle

Proof. For the graph $G = (K_{m,n} \times C_r)$ it is shown in the lemma 2.2 that maximum degree is $\Delta = m + 2$.

From the theorem 3.1 we got that $\lambda_{3,2,1}(K_{m,n} \times C_r) = 2(m+n)$. As we know

$$\begin{aligned} \Delta^3 + 2\Delta = (m+2)^3 + 2(m+2) &> (m+2)^2 \\ &> m^2 + 2m \quad \text{as } m, n > 0 \\ &\geq 2n + 2m \quad \text{as } m > n \text{ so } m^2 \geq 2n \\ &\geq 2(m+n). \end{aligned}$$

Hence the proof. \square

Case:2 For $n > m$ and for any length of cycle

Proof. For the graph $G = (K_{m,n} \times C_r)$ it is shown in the lemma 2.2 that maximum degree is $\Delta = n+2$. From the theorem 3.1 we got that $\lambda_{3,2,1}(K_{m,n} \times C_r) = 2(m+n)$. As we know

$$\begin{aligned} \Delta^3 + 2\Delta = (n+2)^3 + 2(n+2) &> (n+2)^2 \\ &> n^2 + 2n \quad \text{as } m, n > 0 \\ &\geq 2m + 2n \quad \text{as } n > m \text{ so } n^2 \geq 2m \\ &\geq 2(m+n). \end{aligned}$$

Hence the proof. \square

Theorem 3.6. For any sub-graph H of the graph $G = (K_{m,n} \times C_r)$, $\lambda_{3,2,1}(H) \leq 2(m+n)$.

Proof. As we have the span of the graph $G = (K_{m,n} \times C_r)$ is $\lambda_{3,2,1}(K_{m,n} \times C_r) = 2(m+n)$. We consider the graph H , where $H \in \text{subgraph}(G)$ that means H is one of the sub-graph from the sub-graph set of G , so according to lemma 2.3 we can conclude that $\lambda_{3,2,1}(H) \leq 2(m+n)$.

Hence the proof. \square

4 Conclusion

To reduce noise in the form of interference we use $L(3,2,1)$ labeling and apply it to the graph Cartesian product between complete bipartite graph with path and cycle. We developed two efficient algorithms with quadratic time complexity to achieved the labeling which is equal to the $L(3,2,1)$ labeling of the graph $G = K_{m,n}$ i.e $\lambda_{3,2,1}(K_{m,n} \times C_r) = \lambda_{3,2,1}(K_{m,n} \times P_r) = \lambda_{3,2,1}(K_{m,n}) = 2(m+n)$. We also analyzed both the algorithms with suitable examples. In future we are thinking to explore $L(4,3,2,1)$ labeling on some complex structure of graph.

Acknowledgement: The work is supported by the Department of Science and Technology, New Delhi, India, Ref. No.SB/S4/MS: 894/14.

References

- [1] Hale, W. K., Frequency assignment: Theory and applications, Proc. IEEE. 68 (1980) 1497-1514.
- [2] Griggs, J., Yeh, R.K.: Labeling graphs with a condition at distance two. SIAM J. Discret. Math. 5, 586-595 (1992).
- [3] Havet, F., Reed, B., Sereni, J.S.: $L(2,1)$ -labeling of graphs, In: Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, SIAM 621-630 (2008).
- [4] Gonçalves, D.: On the $L(d, 1)$ -labelling of graphs. Discret. Math. 308, 1405-1414 (2008).
- [5] Chang, G.J., Kuo, D.: The $L(2,1)$ -labeling on graphs. SIAM J. Discrete Math. 9, 309-316 (1996).

- [6] Paul, S., Pal, M., Pal, A.: $L(2,1)$ -labeling of permutation and bipartite permutation graphs. *Math. Comput. Sci.* doi:10.1007/s11786-014-0180-2.
- [7] J. Clipperton, J. Gehrtz, Z. Szaniszló, D. Torkornoo, $L(3,2,1)$ -labeling of Simple Graphs, VERUM, Valparaiso University, (2006).
- [8] M.L. Chia, D. Qua, H. Liao, C. Yang, R.K. Yea, $L(3,2,1)$ -labeling of graphs, *Taiwanese J. Math.* 15 (6) (2011) 2439-2457.
- [9] J. Liu, Z. Shao, The $L(3,2,1)$ -labeling problem on graphs, *Math. Appl.* 17 (4) (2004) 596-602.
- [10] J. Clipperton, $L(d, 2, 1)$ -labeling of simple graphs, *Math J.* 9 (2008) 1-11.
- [11] B.M. Kim, W. Hwang, B.C. Song, $L(3,2,1)$ -labeling for product of a complete graph and cycle, *Taiwanese J. Math.* (2014) <http://dx.doi.org/10.11650/tjm.18.2014.4632>.
- [12] J. Clipperton, $L(4, 3, 2, 1)$ -labeling of simple graphs, *Appl. Math. Sci.* 1(2) (2011) 95-102.
- [13] Ghosh, S., Pal, A.: $L(3,1)$ -Labeling of Some Simple Graphs. *Advanced Modeling and Optimization*, 18(2), 243-248(2016).
- [14] Ghosh, S., Paul, S., Pal, A.: $L(2,1)$ -Labeling of Cartesian Product of Complete Bipartite Graph and Path. *Journal of Informatics and Mathematical Sciences*, 9(3), 675-684(2017).
- [15] Ghosh, S., Sarkar, P., Pal, A.: Exact Algorithm for $L(2,1)$ Labeling of Cartesian Product Between Complete Bipartite Graph and Cycle. *Advances in Intelligent Systems and Computing. Harmony Search and Nature Inspired Optimization Algorithms*, 741, 10.1007/978-981-13-0761-4(2019).
- [16] Ghosh, S., Pal, A.: Signed Product and Total Signed Product Cordial Labeling of Cartesian Product Between Balanced Bipartite Graph and Path. *Advanced Computational and Communication Paradigms. Advances in Intelligent Systems and Computing*, Springer, 704, 10.1007/978-981-10-8237-5-50(2018).
- [17] Sk. Amanathulla, M. Pal, $L(3,2,1)$ - and $L(4, 3, 2, 1)$ -labeling problems on circular-arc graphs, *Int. J. Control Theory Appl.* 9 (34) (2016) 869-884.
- [18] Sk. Amanathulla, M. Pal, $L(3,2,1)$ -labeling problems on permutation graphs, *Transylv. Rev.* 25 (14) (2017) 3939-3953.