Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

Edge Sum Divisor Cordial Labeling of Some Graphs with Python Implementation

[1]A. Anto Cathrin Aanisha, [2]R. Manoharan
[1]Research Scholar, Sathyabama Institute of Science and Technology, Chennai, India.
[2]Assistant Professor, Sathyabama Institute of Science and Technology, Chennai, India.

Abstract:- Let $\Omega = (W(\Omega), F(\Omega))$ be a graph that has neither loop nor multiple edges, where $W(\Omega)$ represents the node set and $F(\Omega)$ represents the line set and let $h: F(\Omega) \rightarrow \{1, 2, \ldots |F(\Omega)|\}$ be a bijection. For each node u, give it a label of 1 if $2 \mid h(b_1) + h(b_2) + \ldots + h(b_s)$ and 0 if it doesn't where $b_1, b_2, \ldots b_s$ are lines that are incident with the node u. If the difference between nodes categorized 0 and 1 is less than or equal to 1, the function h is called ESDC labelling. An ESDC graph is one that has the ESDC labeling. In this paper, we prove the circular ladder graph CL_s when s is even, the subdivision of the star $S(K_{1,s})$, the bistar graph $B_{s.s}$ when s is odd and the star graph $K_{1,s}$ when s is even are ESDC graphs. Also in this paper, we investigate an edge sum divisor cordial labeling of the above graphs using python language.

Keywords*:* SDC graph, ESDC graph, the circular ladder graph CL_s, the subdivision of the star $S(K_{1,s})$, the bistar graph $B_{s.s}$, the star graph $K_{1,s}$.

1. Introduction

A study of graphs is referred to as graph theory and graph. Beyond mathematics, graph theory finds use in proving mathematical theorems and models and solving issues in computer science, biology, chemistry, digital image processing, website design, software engineering, operations research, and software engineering, among other scientific and technological domains. [1]. Graph labeling is a fundamental aspect of graph theory and numerous applications employ graph theory, including database management, circuit design, x-ray crystallography, radar, astronomy, and communication network addressing. [2]. Electrical networks can benefit from graph labeling. As an example of a graph representation, consider a circuit network. This form of graph shows the current flowing through the circuit as well as the current determined by the series and parallel connections of the resistors [3]. Graph labeling is a function that turns a set of points (vertices or lines) into positive numbers [4]. In the year 1987, Cahit was the one who first presented the idea of cordial labeling [5]. Divisor cordial labeling or DC labeling, prime cordial labeling or PC labeling, total cordial labeling or TC labeling, Fibonacci cordial labeling or FC labeling, and other types of cordial labeling are all examples of the expansion of cordial labeling [6]. Sum divisor cordial labeling is an extended result on DC labeling. Sum divisor cordial labeling is also known as SDC labeling. The idea of SDC labeling was initially proposed by [7] and at first, they demonstrated that the path, the comb, the star, the full bipartite, the bistar, the crown, the flower, the gear, and the subdivision of the star are all examples of SDC graphs. Edge sum divisor cordial labeling is an extended result on sum divisor cordial labeling [8]. It is also known as ESDC labeling. They demonstrated that path, cycle, fan, wheel, closed helm, friendship graph and flower graph are all examples of ESDC graph. In the realm of graph theory, Python is a popular programming language utilized for various tasks, including graph labeling. Graph labeling involves assigning labels or numerical values to the vertices, edges, or other elements of a graph according to certain rules or properties [9] explored labeling techniques applied to the Grotzsch graph and Peterson Graph using Python programming. Through extensive analysis, they have discovered over a thousand unique Graceful and Odd Even Graceful labeling patterns. [10] presented Python programming techniques for generating the vertices of the Tadpole graph. In this paper, we prove the circular ladder graph CL_s when s is even, the subdivision of the star $S(K_{1,s})$, the bistar graph $B_{s.s}$ when s is odd and the star graph $K_{1,s}$ when s is even are ESDC graphs. Also, we investigate an edge sum divisor cordial labeling of the above graphs using python language. SDC definition is follows: Let $\Omega = (W(\Omega), F(\Omega))$ be a graph which is having neither loop nor multiple edges, where $W(\Omega)$ represent node set and $F(\Omega)$

represent line set and let h: $W(\Omega) \rightarrow \{1, 2, \ldots |W(\Omega)|\}$ be a bijection. For each line x, give it a label of 1 if $2/(h(x)+h(y))$ and 0 if not. If the difference between lines categorized 0 and 1 is less than or equal to 1, the function h is called SDC labelling. A SDC graph is one that has the SDC labeling [7]. ESDC definition is follows: Let $\Omega = (W(\Omega), F(\Omega))$ be a graph which is having neither loop nor multiple edges, where $W(\Omega)$ represent node set and $F(\Omega)$ represent line set and let h: $F(\Omega) \rightarrow \{1, 2, \ldots |F(\Omega)|\}$ be a bijection. For each node u, give it a label of 1 if $2 \mid h(b_1)+ h(b_2)+...+ h(b_s)$ and 0 if it doesn't where $b_1, b_2, \ldots b_s$ are lines that are incident with the node u. If the difference between nodes categorized 0 and 1 is less than equal to 1, the function h is called ESDC labelling. A ESDC graph is one that has the ESDC labeling [8]. All graphs discussed here are neither loop nor multiple edges, finite number of vertices and edges, connected, and undirected. The definition of the circular ladder graph CL_s is $P_2 \Theta C_s$ [11]. The subdivision of star $S(K_{1,s})$ is created by adding a line of degree one to each node of $K_{1,s}$, excluding the end node [7]. Apex vertices of two copies of $K_{1,s}$ by an edge. The graph that is created by connecting the apex vertices of two copies of $K_{1,s}$ by means of an edge is referred to as the bistar $B_{s,s}$.

2. Objectives

The paper aims to prove ESDC properties in CL_s, $S(K_{1,s})$, $B_{s.s}$, and $K_{1,s}$ graphs. It also investigates ESDC labeling on these graphs using Python, exploring both theoretical and computational aspects.

3. Main Results:

Theorem: 3.1

The circular ladder CL_s is an ESDC graph for $s \geq 4$ and s is even.

Proof:

Let $\Omega = CL_s$ where s is even and $s \geq 4$

Let $x_1, x_2, x_3 \ldots x_{2s}$ be the nodes of Ω and $f_1, f_2, f_3 \ldots f_{3s}$ be the lines of Ω such that

$f_k = x_k x_{k+1}$; $1 \leq k \leq s-1$

$f_s = x_1 x_s$,

$f_{s+k} = x_k x_{s+k}$; $1 \leq k \leq s$

$f_{2s+k} = x_{s+k} x_{s+1+k}$; $1 \leq k \leq s-1$

$f_{3s} = x_{2s} x_{s+1}$

Also $|W(\Omega)| = 2s$ and $|F(\Omega)| = 3s$

Define h: $F(\Omega) \rightarrow \{1, 2, \ldots |F(\Omega)|\}$ as follows:

$h(f_k) = 3k$; $1 \leq k \leq s$

$h(f_{s+k}) = 3k-1$; $1 \leq k \leq s$

$h(f_{2s+k}) = 3k-2$; $1 \leq k \leq s$

Then induced nodes labels are

$h^*(x_{2k-1}) = 0$; $1 \leq k \leq s$

$h^*(x_{2k}) = 1$; $1 \leq k \leq s$

In light of the aforementioned labeling, we have, we have $x_h(1) = x_h(0) = s$

Here $x_h(1)$ represents the number of nodes labeled with 1 and $x_h(0)$ represents the number of nodes labeled with 0 among all nodes.

Thus $| x_h(1) - x_h(1)| = |s-s| = 0 \leq 1$

Hence the circular ladder CL_s is an ESDC graph for $s \geq 4$ and s is even.

Example: 3.2

An ESDC labeling of the circular ladder CL_6 is shown in Figure 1

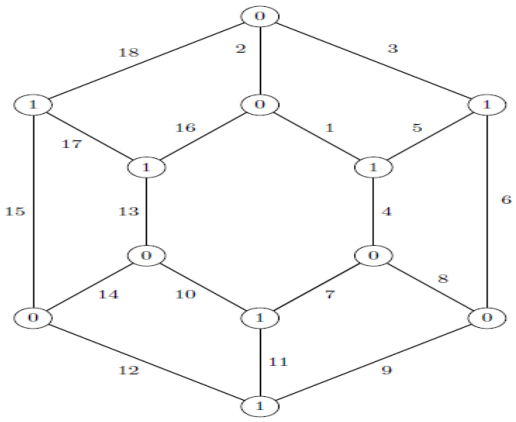

Figure 1

From figure 1, $|x_h(0)-x_h(1)|= |6-6|= 0\leq 1$.

So, we conclude that the circular ladder CL_6 is having ESDC labeling.

Hence the circular ladder CL_6 is an ESDC graph.

Theorem: 3.3

The graph subdivision of the star $S(K_{1,s})$ is an ESDC graph.

Proof:

Let $\Omega= S(K_{1,s})$

Let $x, x_1, x_2, \ldots x_{2s}$ be the nodes of Ω and $f_1, f_2, \ldots f_{2s}$ be the lines of Ω such that

$f_k= xx_k$; $1\leq k\leq s$

$f_{s+k}= x_k\, x_{k+s}$; $1\leq k\leq s$

Also $|W(\Omega)|= 2s+1$ and $|F(\Omega)|= 2s$

Define $h: F(\Omega)\rightarrow\{1, 2, 3 \ldots |F(\Omega)|\}$ as follows:

$h(f_k)= 2k- 1$; $1\leq k\leq s$

$h(f_{s+k})= 2k$; $1\leq k\leq s$

Then induced nodes labels are

$$h^*(x) = \begin{cases} 0 & if\ x\ is\ odd \\ 1 & if\ x\ is\ even \end{cases}$$

$h^*(x_k)= 0$; $1\leq k\leq s$

$h^*(x_{s+k})= 1$; $1\leq k\leq s$

In light of the aforementioned labeling, we have

$$x_h(1) = \begin{cases} s & if\ s\ is\ odd \\ s+1 & if\ s\ is\ even \end{cases}$$

$$x_h(0) = \begin{cases} s+1 & if\ s\ is\ odd \\ s & if\ s\ is\ even \end{cases}$$

Thus $|x_h(0)- x_h(1)|\leq 1$

Hence the graph subdivision of the star $S(K_{1,s})$ is an ESDC graph.

Example: 3.4

An ESDC labeling of subdivision of the star $S(K_{1,3})$ is shown in Figure 2

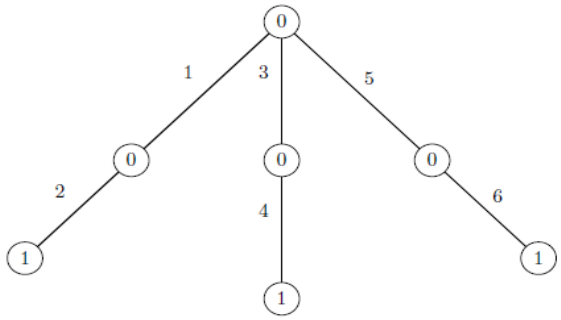

Figure 2

From figure 2, $|x_h(0)-x_h(1)|= |4-3|= 1\leq 1$.

So, we conclude that the subdivision of the star $S(K_{1,3})$ is having ESDC labeling.

Hence the subdivision of the star $S(K_{1,3})$ is an ESDC graph.

Theorem: 3.5

The bistar graph $B_{s,s}$ is an ESDC graph for s is odd.

Proof:

Let $\Omega = B_{s,s}$, where s is odd

Let $W(\Omega)= \{x, y, x_k, y_k; 1\leq k\leq s\}$ be the nodes and $F(\Omega)= \{f, f_1, f_2, f_3 \ldots f_{2s}\}$ be the edges of Ω such that

$f_k= xx_k; 1\leq k\leq s$

$f_{s+k}= yy_k; 1\leq k\leq s$

$f=xy$

Also $|W(\Omega)|= 2s+2$ and $|F(\Omega)|= 2s+1$

Define $h: F(\Omega)\rightarrow \{1, 2, 3 \ldots |f(\Omega)|\}$ as follows:

$h(f_k)= 2k; 1\leq k\leq s$

$h(f_{s+k})= 2k+1; 1\leq k\leq s$

$h(f)= 1;$

Then induced nodes labels are

$h^*(x)= 0$

$h^*(y)= 1$

$h^*(x_k)= 1; 1\leq k\leq s$

$h^*(y_k)= 0; 1\leq k\leq s$

In light of the aforementioned labeling, we have

$x_h(1)= x_h(0)= s+1$

Here $x_h(1)$ represents the number of nodes labeled with 1 and $x_h(0)$ represents the number of edges labeled with 0 among all nodes.

Thus, | $x_h(0)$- $x_h(1)$|≤ 1

Hence, the bistar graph $B_{s,s}$ is an ESDC graph for s is odd.

Example: 3.6

An ESDC labeling of bistar graph $B_{3,3}$ is shown in Figure 3

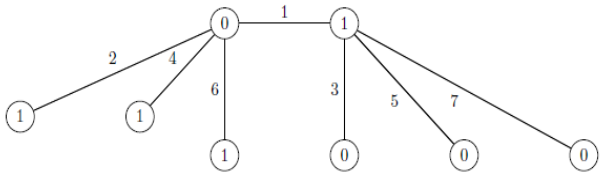

Figure 3

From figure 3, $|x_h(0)-x_h(1)|= |4-4|= 0\leq 1$.

So, we conclude that the bistar graph $B_{3,3}$ is having ESDC labeling.

Hence the bistar graph $B_{3,3}$ is an ESDC graph.

Theorem: 3.7

The star graph $K_{1,s}$ is an ESDC graph for all s is even.

Proof:

Let Ω= $K_{1,s}$, where s is even

Let x, x_1, x_2, x_3 . . . x_s be the nodes of Ω and f_1, f_2, f_3 . . . f_s be the lines of Ω such that

$f_k = xx_k$; 1≤ k≤ s

Also |W(Ω)| = s+1 and |F(Ω)|= s

Define h: F(Ω)→ {1, 2, 3 . . .|F(Ω)|} as follows:

$h(f_k)= k$; 1≤ k≤ s

Then induced nodes labels are

$h^*(x)= = \begin{cases} 0 & if\ s=4p-2, p\epsilon N \\ 1 & if\ s=4p, p\epsilon N \end{cases}$

$h^*(x_{2k})= 1$; $1\leq k\leq \frac{s}{2}$

$h^*(x_{2k-1})= 0$; $1\leq k\leq \frac{s}{2}$

In light of the aforementioned labeling, we have

$x_h(0)= \begin{cases} \frac{s}{2}+1 & if\ s=4p-2, p\epsilon N \\ \frac{s}{2} & if\ s=4p, p\epsilon N \end{cases}$

$x_h(1)= \begin{cases} \frac{s}{2} & if\ s=4p-2,\ p\epsilon N \\ \frac{s}{2}+1 & if\ s=4p, p\epsilon N \end{cases}$

Thus, | $x_h(0)$- $x_h(1)$|≤ 1

Hence, the star graph $K_{1,s}$ is an edge sum divisor cordial graph for all s is even.

Example: 3.8

An ESDC labeling of the star graph $K_{1,3}$ is shown in Figure 4

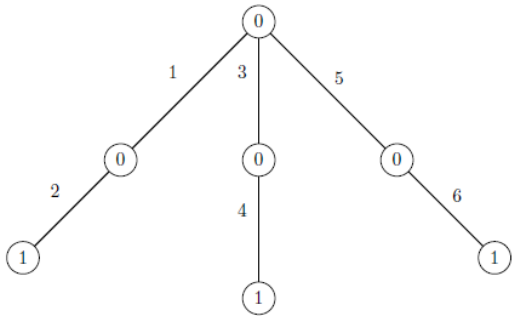

Figure 4

From figure 4, $|x_h(0)-x_h(1)|= |4-3|= 1\leq 1$.

So, we conclude that the star graph $K_{1,3}$ is having ESDC labeling.

Hence the star graph $K_{1,3}$ is an ESDC graph.

4. Exploring edge sum divisor cordial labeling in graphs using Python.

4.1 Python Implementation to Verify Edge Sum Divisor Cordiality of the star graph $K_{1,s}$ for all s is even

```
def star_ESDC(s):
    if s % 2 != 0:
        print("s must be even for the star graph to be an ESDC graph.")
        return

    nodes = ["x"] + ["x" + str(k) for k in range(1, s + 1)]
    edges = [("x", "x" + str(k)) for k in range(1, s + 1)]

    # Define the labeling function
    def label_edges(edge):
        index = edges.index(edge) + 1  # Convert edge index to 1-based index
        return index

    # Label the edges
    labeled_edges = {edge: label_edges(edge) for edge in edges}
    print("Edge labels:", labeled_edges)

    # Define the induced node labels
    induced_nodes_labels = {}
    if s % 4 == 2:
        induced_nodes_labels["x"] = 0
    else:
        induced_nodes_labels["x"] = 1

    for k in range(1, s // 2 + 1):
        induced_nodes_labels["x" + str(2 * k)] = 1
        induced_nodes_labels["x" + str(2 * k - 1)] = 0

    print("Induced node labels:", induced_nodes_labels)

    # Check ESDC property
    num_label_0 = sum(1 for label in induced_nodes_labels.values() if label == 0)
    num_label_1 = sum(1 for label in induced_nodes_labels.values() if label == 1)
    difference = abs(num_label_0 - num_label_1)
    is_ESDC = difference <= 1
    print("Is edge sum divisor cordial:", is_ESDC)


# Test the function with s = 6
star_ESDC(6)
```

```
PS C:\Users\User> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/User/AppData/Local/Packages/PythonSoftwareFoundation.Python.3.12
_qbz5n2kfra8p0/LocalCache/local-packages/Python312/site-packages/matplotlib/pyplot.py
Edge labels: {('x', 'x1'): 1, ('x', 'x2'): 2, ('x', 'x3'): 3, ('x', 'x4'): 4, ('x', 'x5'): 5, ('x', 'x6'): 6}
Induced node labels: {'x': 0, 'x2': 1, 'x1': 0, 'x4': 1, 'x3': 0, 'x6': 1, 'x5': 0}
Is edge sum divisor cordial: True
```

4.2 Python Implementation to Verify Edge Sum Divisor Cordiality of Circular Ladder CLs for s≥ 4 and s is even

```python
def circular_ladder_ESDC(s):
    if s < 4 or s % 2 != 0:
        print("s must be even and greater than or equal to 4.")
        return

    edges = []
    for k in range(1, s):  # Generate edges f(k)
        edges.append(("x(" + str(k) + ")", "x(" + str(k + 1) + ")"))
    edges.append(("x(1)", "x(" + str(s) + ")"))  # Generate edge f(s)
    for k in range(1, s + 1):  # Generate edges f(s+k)
        edges.append(("x(" + str(k) + ")", "x(" + str(s + k) + ")"))
    for k in range(1, s):  # Generate edges f(2s+k)
        edges.append(("x(" + str(s + k) + ")", "x(" + str(s + 1 + k) + ")"))
    edges.append(("x(" + str(2 * s) + ")", "x(" + str(s + 1) + ")"))  # Generate edge f(3s)

    # Define the labeling function
    def label_edges(edge):
        index = edges.index(edge) + 1  # Convert edge index to 1-based index
        if index <= s:
            return 3 * index  # Label for f(k)
        elif s < index <= 2 * s:
            return 3 * (index - s) - 1  # Label for f(s+k)
        else:
            return 3 * (index - 2 * s) - 2  # Label for f(2s+k)

    # Label the edges
    labeled_edges = {edge: label_edges(edge) for edge in edges}
    print("Edge labels:", labeled_edges)

    # Label the induced nodes
    induced_nodes_labels = {}
    for k in range(1, s + 1):  # Generate induced node labels
        induced_nodes_labels["x(" + str(2 * k - 1) + ")"] = 0
        induced_nodes_labels["x(" + str(2 * k) + ")"] = 1
    print("Induced node labels:", induced_nodes_labels)
    # Check ESDC property
    num_label_0 = sum(1 for label in induced_nodes_labels.values() if label == 0)
    num_label_1 = sum(1 for label in induced_nodes_labels.values() if label == 1)
    difference = abs(num_label_0 - num_label_1)
    is_ESDC = difference <= 1

    # Ensure all edges are assigned labels
    for edge in edges:
        if edge not in labeled_edges:
            print(f"Warning: Edge {edge} is not assigned a label.")

    print("Is edge sum divisor cordial:", is_ESDC)


# Test the function with s = 4
circular_ladder_ESDC(4)
```

```
PS C:\Users\User> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/User/AppData/Local/Packages/PythonSoftwareFoundation
.Python.3.12_qbz5n2kfra8p0/LocalCache/local-packages/Python312/site-packages/matplotlib/pyplot.py
Edge labels: {('x(1)', 'x(2)'): 3, ('x(2)', 'x(3)'): 6, ('x(3)', 'x(4)'): 9, ('x(1)', 'x(4)'): 12, ('x(1)', 'x(5)'): 2, ('x(2)', 'x(6)'): 5, ('x(3
)', 'x(7)'): 8, ('x(4)', 'x(8)'): 11, ('x(5)', 'x(6)'): 1, ('x(6)', 'x(7)'): 4, ('x(7)', 'x(8)'): 7, ('x(8)', 'x(5)'): 10}
Induced node labels: {'x(1)': 0, 'x(2)': 1, 'x(3)': 0, 'x(4)': 1, 'x(5)': 0, 'x(6)': 1, 'x(7)': 0, 'x(8)': 1}
Is edge sum divisor cordial: True
```

4.3 Python Implementation to Verify Edge Sum Divisor Cordiality of subdivision of the star $S(K_{1,s})$.

```python
def star_subdivision_ESDC(s):
    if s < 1:
        print("s must be greater than or equal to 1.")
        return

    edges = []
    for k in range(1, s + 1):  # Generate edges f(k)
        edges.append(("x", "x(" + str(k) + ")"))
    for k in range(1, s + 1):  # Generate edges f(s+k)
        edges.append(("x(" + str(k) + ")", "x(" + str(k + s) + ")"))

    # Define the labeling function
    def label_edges(edge):
        index = edges.index(edge) + 1  # Convert edge index to 1-based index
        if index <= s:
            return 2 * index - 1  # Label for f(k)
        else:
            return 2 * (index - s)  # Label for f(s+k)

    # Label the edges
    labeled_edges = {edge: label_edges(edge) for edge in edges}
    print("Edge labels:", labeled_edges)

    # Define the induced node labels
    induced_nodes_labels = {}
    if s % 2 == 0:
        induced_nodes_labels["x"] = 1
    else:
        induced_nodes_labels["x"] = 0
    for k in range(1, s + 1):
        induced_nodes_labels["x(" + str(k) + ")"] = 0
        induced_nodes_labels["x(" + str(k + s) + ")"] = 1

    print("Induced node labels:", induced_nodes_labels)

    # Check ESDC property
    num_label_0 = sum(1 for label in induced_nodes_labels.values() if label == 0)
    num_label_1 = sum(1 for label in induced_nodes_labels.values() if label == 1)
    difference = abs(num_label_0 - num_label_1)
    is_ESDC = difference <= 1
    print("Is edge sum divisor cordial:", is_ESDC)


# Test the function with s = 5
star_subdivision_ESDC(5)
```

```
PS C:\Users\User> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/User/AppData/Local/Packages/PythonSoftwareFoundation
.Python.3.12_qbz5n2kfra8p0/LocalCache/local-packages/Python312/site-packages/matplotlib/pyplot.py
Edge labels: {('x', 'x(1)'): 1, ('x', 'x(2)'): 3, ('x', 'x(3)'): 5, ('x', 'x(4)'): 7, ('x', 'x(5)'): 9, ('x(1)', 'x(6)'): 2, ('x(2)', 'x(7)'): 4,
('x(3)', 'x(8)'): 6, ('x(4)', 'x(9)'): 8, ('x(5)', 'x(10)'): 10}
Induced node labels: {'x': 0, 'x(1)': 0, 'x(6)': 1, 'x(2)': 0, 'x(7)': 1, 'x(3)': 0, 'x(8)': 1, 'x(4)': 0, 'x(9)': 1, 'x(5)': 0, 'x(10)': 1}
Is edge sum divisor cordial: True
```

4.4 Python Implementation to Verify Edge Sum Divisor Cordiality of the bistar graph $B_{s,s}$ for s is odd

```python
def bistar_ESDC(s):
    if s % 2 == 0:
        print("s must be odd for bistar graph to be an ESDC graph.")
        return

    nodes = ["x", "y"] + ["x" + str(k) for k in range(1, s + 1)] + ["y" + str(k) for k in range(1, s + 1)]
    edges = [("x", "x" + str(k)) for k in range(1, s + 1)] + [("y", "y" + str(k)) for k in range(1, s + 1)] + [("x", "y")]

    # Define the labeling function
    def label_edges(edge):
        index = edges.index(edge) + 1  # Convert edge index to 1-based index
        if index <= s:
            return 2 * index  # Label for fk
        elif index == 2 * s + 1:
            return 1  # Label for f
        else:
            return 2 * (index - s) + 1  # Label for fs+k

    # Label the edges
    labeled_edges = {edge: label_edges(edge) for edge in edges}
    print("Edge labels:", labeled_edges)

    # Define the induced node labels
    induced_nodes_labels = {}
    induced_nodes_labels["x"] = 0
    induced_nodes_labels["y"] = 1
    for k in range(1, s + 1):
        induced_nodes_labels["x" + str(k)] = 1
        induced_nodes_labels["y" + str(k)] = 0

    print("Induced node labels:", induced_nodes_labels)

    # Check ESDC property
    num_label_0 = sum(1 for label in induced_nodes_labels.values() if label == 0)
    num_label_1 = sum(1 for label in induced_nodes_labels.values() if label == 1)
    difference = abs(num_label_0 - num_label_1)
    is_ESDC = difference <= 1
    print("Is edge sum divisor cordial:", is_ESDC)


# Test the function with s = 3
bistar_ESDC(3)
```

```
PS C:\Users\User> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/User/AppData/Local/Packages/PythonSoftwareFoundation.Python.3.12
_qbz5n2kfra8p0/LocalCache/local-packages/Python312/site-packages/matplotlib/pyplot.py
Edge labels: {('x', 'x1'): 2, ('x', 'x2'): 4, ('x', 'x3'): 6, ('y', 'y1'): 3, ('y', 'y2'): 5, ('y', 'y3'): 7, ('x', 'y'): 1}
Induced node labels: {'x': 0, 'y': 1, 'x1': 1, 'y1': 0, 'x2': 1, 'y2': 0, 'x3': 1, 'y3': 0}
Is edge sum divisor cordial: True
```

5. Discussion

The discussion will talk about what it means to prove that certain graphs (CL_s, $S(K_{1,s})$, $B_{s,s}$, and $K_{1,s}$) can have ESDC labeling. This means we'll understand more about how these graphs can be labeled in a balanced way. We'll look at why it's important to balance the number of nodes labeled 0 and 1. The paper will also discuss how we used Python to find these labelings, and whether the methods we used are efficient and easy to use. We'll consider where this type of labeling could be useful and what its limitations are. We'll compare it to other labeling methods

to show why ESDC labeling might be better. Finally, we'll suggest future research ideas, like trying ESDC labeling on different graphs and improving our techniques.

Refrences

[1] R. D. More, A. S. Patil, D. S. Dandwate, U. J. Tupe, and C. Science, "A Literature Review on Applications of Graph Theory in Various Fields," vol. 6, no. 1, pp. 1–8, 2021.

[2] N. La. Prasanna, K. Sravanthi, and N. Sudhakar, "Applications of Graph Labeling in Communication Networks," Orient. J. Comput. Sci. Technol., vol. 7, no. 1, pp. 139–145, 2014, [Online]. Available: www.computerscijournal.org

[3] G. Dhanalakshmi, D. A. Emiltet, G. Dhanalakshmi, D. A. Emiltet, and C. Network, "A view point on Applications of Graph Labeling in Communication Networks," vol. 11, pp. 682–690, 2022.

[4] S. Sadawarte and S. Srivastav, "Signed product cordial labeling of some pan graphs," Mater. Today Proc., vol. 57, pp. 2307–2310, 2022, doi: 10.1016/j.matpr.2022.01.097

[5] R. Ponraj, M. Sivakumar, and M. Sundaram, "Mean Cordial Labeling of Graphs," Open J. Discret. Math., vol. 02, no. 04, pp. 145–148, 2012, doi: 10.4236/ojdm.2012.24029.

[6] A. Raheem, M. Javaid, M. Numan, and R. Hasni, "Sum divisor cordial labeling of disjoint union of paths and subdivided star," J. Discret. Math. Sci. Cryptogr., vol. 25, no. 8, pp. 2467–2478, 2022, doi: 10.1080/09720529.2020.1864935.

[7] A. Lourdusamy and F. Patrick, "Sum divisor cordial graphs," Proyecciones, vol. 35, no. 1, pp. 119–136, 2016, doi: 10.4067/S0716-09172016000100008.

[8] G. Vijayalakshmi, M. M. Sheriff, P. L. R. Raj, T. Nadu, and H. K. Rowther, "Journal of Clinical Otorhinolaryngology , Head , and Neck Surgery," no. December 2022, pp. 407–417.

[9] J. A. Gadhiya and R. Solanki, "Labeling of Some Graphs Using Python Programming," vol. 20, no. 17, pp. 2288–2294, 2022, doi: 10.48047/Nq.2022.20.17.Nq880294.

[10] A. U. Maheswari, "Arithmetic Number Labelling for Banana tree , Olive tree , Shrub , Jelly fish , Tadpole graphs," vol. 7, no. 7, pp. 264–277, 2022.

[11] C. M. Barasara and Y. B. Thakkar, "DIVISOR CORDIAL LABELING FOR LADDERS AND TOTAL GRAPH OF SOME GRAPHS," vol. 21, no. 7, pp. 3577–3594, 2022.