International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

# Computational Approaches to Mean Based Cordial Labeling of Graphs Using Python

Jeena R.V[1] and Asha S[2]

**Submitted:**02/12/2023 **Revised:**05/01/2024 **Accepted:**15/01/2024

**Abstract:** Graph theory has developed quickly in recent years. This is due to its mathematical significance as well as the numerous applications that stem from it, which include coding theory, radar, astronomy, circuit design, communication design, X-ray crystallography, and others. Labeled graphs may be useful in a wide range of applications. The wide variety of applications for labeled graphs leads us to research different types of mean cordial labeling of graphs.

***Keywords****: applications, crystallography, numerous*

**Subject Classification**: 05C78

## 1. INTRODUCTION

Graphs $G$ considered here are simple, finite, connected and undirected. The vertex set and edge set of a graph $G$ are $V(G)$ and $E(G)$ respectively. The concept of cordial labeling was introduced by Cahit in the year 1987. Mean cordial labeling of graphs was introduced by Ponraj. R and Somasundaram. S. The concept of prime mean labeling was introduced by Palani. K. The concept of difference mean cordial labeling was introduced by Savithri. R and Shenbega Devi. S. Python is a high-level, interpreted programming language with a developer-friendly design and a focus on clarity and readability. The distinctive use of indentation to mark code blocks enhances code readability and creates a cleaner program appearance. Python, developed by Guido van Rossum in the late 1980s and introduced in 1991, was intended to replace the ABC programming language.

### PRELIMINARIES

**Definition 1.1** Let $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ be a graph with $\mathrm{p}$ vertices and $\mathrm{q}$ edges. A graph $\mathrm{G}$ is said to be a prime difference mean cordial labeling if there exists an injective function $f: \mathrm{V(G)} \to \{0,1,2,\dots,\mathrm{q}\}$ such that the induced edge labeling $f^*: \mathrm{E(G)} \to \{0,1\}$ defined by

$$f^*(e = uv) = \begin{cases} 1 & if \;\; f(u) - f(v) \equiv 0(mod2) \\ 0 & otherwise \end{cases}$$

satisfying the condition that for every $v \in V(G)$ with $deg(v) \geq 1$, $S_v = \sum\{f^*(e = uv)/uv \in E(G)\}$ is 1 or prime and $\left|e_f(i) - e_f(j)\right| \leq 1, \forall\, i,j \in \{0,1\}$ where $e_f(x)$ denote the number of edges labeled with $x$. A graph with prime difference mean cordial labeling is called prime difference mean cordial graph.

**Definition 1.2** Let $G = (V, E)$ be a graph with $p$ vertices and $q$ edges. A graph $G$ is said to be a prime mean cordial graph if there exists an injective function $f\colon V(G) \to \{0,1,2, \dots, q\}$ such that the induced edge labeling $f^*\colon E(G) \to \{0,1\}$ defined by

$$f^*(e = uv)$$
$$\begin{cases} 1 & if \; \left\lceil \frac{f(u)+f(v)}{2} \right\rceil \;\; is\ odd \\ 0 & \qquad otherwise \end{cases}$$

[1]Research Scholar (Full time), Register Number: 21113112092015,
Research Department of Mathematics, Nesamony Memorial Christian College, Marthandam Kanniyakumari District, Tamil Nadu, India, Affiliated to Manonmaniam Sundaranar University, Tirunelveli- 627 012, Abishekapatti, Tamil Nadu, India.
E-mail: vrjeena@gmail.com
[2]Associate Professor, Research Department of Mathematics, Nesamony Memorial
Christian College, Marthandam, Kanniyakumari District, Tamil Nadu, India, Affiliated to Manonmaniam Sundaranar University, Tirunelveli- 627 012, Abishekapatti,
Tamil Nadu, India.
E-mail: ashanmcc@gmail.com

satisfying the condition that for every $v \in V(G)$ with $deg(v) \geq 1$, $S_v = \sum\{f^*(e = uv) / uv \in E(G)\}$ is prime or 1 and $|e_f(i) - e_f(\text{j})| \leq 1, \forall\ i, \text{j} \in \{0,1\}$ where $e_f(x)$ denote the number of edges labeled with $x$. A graph with prime mean cordial labeling is called prime mean cordial graph.

**Definition 1.3** Let $G = (V, E)$ be a graph and $f$ be a mapping from $V(G) \rightarrow \{0,1,2\}$. For each edge $uv$ of $G$ assign the label $\left\lceil \sqrt[3]{\frac{f(u)^2+f(v)^2}{2}} \right\rceil$. Then $f$ is a called $a$ cube root square mean cordial labeling if $|v_f(i) - v_f(j)| \leq 1$ and $|e_f(i) - e_f(j)| \leq 1$, where $v_f(x)$ and $e_f(x)$ denote the number of vertices and edges labeled with $x, x \in \{0,1,2\}$ respectively. A graph with cube root square mean cordial labeling is called cube root square mean cordial graph.

## MAIN RESULTS

### 1. Prime Difference Mean Cordial Labeling of Graphs Using Python Programming

The programming application for finding the prime difference mean cordial labeling of cycle graph using python is given below. This python programming calculates the prime mean cordial labeling of cycle graph for all $n \geq 3$.

```
import networkx as nx
import matplotlib.pyplot as plt
import numpy as np
#Generate node values using the given formula
def generate_f_values(n):
return [(i - 1) // 2 if i % 2 == 1 else (n + i - 2) // 2 for i in range(1, n + 1)]
# Plot a compact cycle graph
def plot_cycle_graph(n):
G = nx.cycle_graph(n)
node_values = generate_f_values(n)
f = {i: node_values[i] for i in range(n)}
f_star = {(u, v): 1 if abs(f[u] - f[v]) % 2 == 1
else 0 for u, v in G.edges()}
node_labels = {i: f"$v_{{{i+1}}}\\{f[i]}$" for i in range(n)}
pos = nx.circular_layout(G)
for k in pos:
pos[k] *= 0.65  # Shrink node positions
plt.figure(figsize=(5.5, 5))  # Small
nx.draw(G, pos, with_labels=False, node_size=50, node_color='black')
# Draw node labels
for i, (x, y) in pos.items():
plt.text(x * 1.12, y * 1.12, node_labels[i],
fontsize=11, fontweight='bold',
ha='center', va='center', color='black')
# Draw edge labels
for (u, v), label in f_star.items():
x1, y1 = pos[u]
x2, y2 = pos[v]
mx, my = (x1 + x2) / 2, (y1 + y2) / 2
dx, dy = x2 - x1, y2 - y1
perp = np.array([-dy, dx]) / np.linalg.norm([dx, dy]) * 0.025
plt.text(mx + perp[0], my + perp[1], str(label),
fontsize=10, color='black', ha='center', va='center')
plt.axis('off')
plt.tight_layout()
plt.show()
plot_cycle_graph(10)
```

The application developed using python programming language was tested and the given figure illustrates the sample output interface.

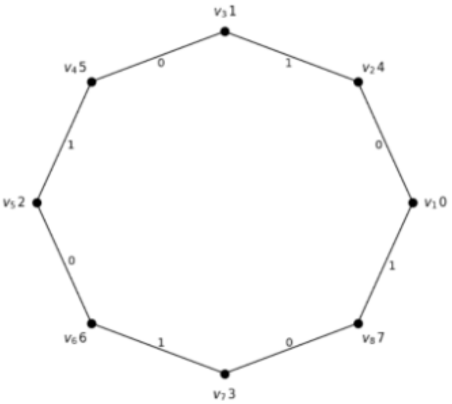

**Figure 1.** Prime difference mean cordial labeling of cycle graph $C_8$

## 2. Prime Mean Cordial Labeling of Graphs Using Python Programming

The programming application for finding the prime mean cordial labeling of brush graph using python is given below. This python programming calculates the prime mean cordial labeling of brush graph for all $n \geq 2$.

```
import networkx as nx
import matplotlib.pyplot as plt
import math
# Define number of pairs
n = 6
# Create graph
G = nx.Graph()
# Create node list
nodes = list(range(2 * n))
G.add_nodes_from(nodes)
# Define f_values correctly as per formula
def f_values(node):
i = node // 2 + 1
if node % 2 == 0:
return 2 * i - 2
else:
return 2 * i - 1
# Define edge function f(u, v)
def f(u, v):
return 1 if math.ceil((f_values(u) + f_values(v)) / 2) % 2 != 0 else 0
# Create vertical edges (comb teeth)
vertical_edges = [(2*i, 2*i+1) for i in range(n)]
# Create horizontal edges (spine)
horizontal_edges = [(2*i+1, 2*i+3) for i in range(n-1)]
# Add edges to graph
for u, v in vertical_edges + horizontal_edges:
G.add_edge(u, v, weight=f(u, v))
# Position nodes: top row y=1 (for v), bottom row y=0 (for u)
pos = {}
for i in range(n):
pos[2*i] = (i, 1)
pos[2*i+1] = (i, 0)
# Plot graph
plt.figure(figsize=(n, 3))
nx.draw(G, pos, with_labels=False, node_size=50, node_color='black')
label_gap = 0.05
# Draw node labels: subscripts + function values
for node, (x, y) in pos.items():
i = node // 2 + 1
val = f_values(node)
if node % 2 == 0:
label = f'$v_{{{i}}}$ {val}'
plt.text(x, y + label_gap, label, fontsize=10, ha='center', va='center')
else:
label = f'$u_{{{i}}}$ {val}'
```

```
plt.text(x, y - label_gap, label, fontsize=10, ha='center', va='center')
# Draw edge function values
for u, v in vertical_edges:
x_mid = (pos[u][0] + pos[v][0]) / 2 - 0.07
y_mid = (pos[u][1] + pos[v][1]) / 2
plt.text(x_mid, y_mid, str(f(u, v)), fontsize=9, ha='center', color='black')
# Horizontal edges
for u, v in horizontal_edges:
x_mid = (pos[u][0] + pos[v][0]) / 2
y_mid = (pos[u][1] + pos[v][1]) / 2 + 0.03
plt.text(x_mid, y_mid, str(f(u, v)), fontsize=9, ha='center', color='black')
plt.axis('off')
plt.show()
```

The application developed using python programming language was tested and the given figure illustrates the sample output interface.

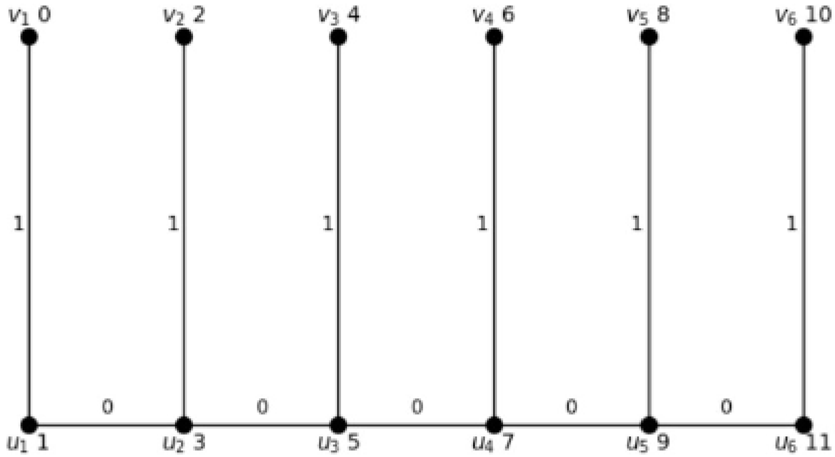

**Figure 2.** Prime mean cordial labeling of brush graph $B_6$

## 3. Cube Root Square Mean Cordial Labeling of Graphs Using Python Programming

The programming application for finding the cube root square mean cordial labeling of path graph using python is given below. This python programme calculates the cube root square mean cordial labeling of path graph for all $n \geq 2$.

```
import matplotlib.pyplot as plt
import networkx as nx
import numpy as np
import math
# Define cordial function
def cordial_function(u, v):
return math.ceil(((u**2 + v**2) / 2)**(1/3))
# Generalized node value generator
def generate_cordial_graph(n):
rem = n % 3
t = n // 3
node_values = []
for i in range(1, n+1):
if rem == 0:
if i <= t:
node_values.append(0)
elif i <= 2*t:
node_values.append(1)
else:
node_values.append(2)
elif rem == 1:
if i <= t+1:
```

```
node_values.append(0)
elif i <= 2*t+1:
node_values.append(1)
else:
node_values.append(2)
elif rem == 2:
if i <= t+1:
node_values.append(0)
elif i <= 2*t+2:
node_values.append(1)
else:
node_values.append(2)
return node_values
# Input: total number of nodes
n = 8
# Generate node values
node_values = generate_cordial_graph(n)
# Create graph
G = nx.path_graph(n)
# Edge values
edge_values =[cordial_function(node_values[i],node_values[i+1]) for i in range(n-1)]
# Positions
pos = {i: (i, 0) for i in range(n)}
# Plot
plt.figure(figsize=(max(7, n/3), 1))
nx.draw(G, pos, with_labels=False, node_color='black', node_size=50)
# Edge function values
for i, value in enumerate(edge_values):
plt.text(i + 0.5, 0.009, str(value), ha='center', fontsize=9)
# Node labels
for node, (x, y) in pos.items():
label = f"$u_{{{node}}}$ {node_values[node]}"
plt.text(x, y - 0.025, label, ha='center', fontsize=9)
plt.axis('off')
plt.tight_layout()
plt.show()
```

The application developed using python programming language was tested and the given figure illustrates the sample output interface.

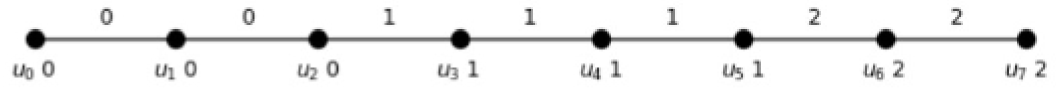

**Figure 3.** Cube root square mean cordial labeling of path graph $P_8$

## Conclusion

In this paper, prime difference mean cordial labeling, prime mean cordial labeling, cube root square mean cordial labeling of graphs using python were found out.

## References

[1] S. Asha, Jeena. R. V, Prime Difference Mean Cordial Labeling of Path and Cycle Graphs, Journal of Computational Analysis and Applications , vol.33, pp. 636-643, 2024.

[2] S. Asha, Jeena. R. V, Cube Root Square Mean Cordial Labeling Of Some Standard Graphs, African Journal of Biomedical and Research, vol.27(6s), pp.572-579, 2024

[3] A. Bondy and U. S. R. Murty, Graph Theory, Springer (2008).

[4] Bondy and V. Chvatal, A method in Graph Theory, Discreate Math., 15 (1976) 111-136.

[5] Cahit.I, Cordial Graphs: A Weaker Version of Graceful and Harmonious Graphs, Ars Combinatoria, vol.23, No.3, pp.201-207, 1987.

[6] J.A.Gallian, A Dynamic Survey of Graph Labeling, The Electronic Journal of Combinotorics, 1(2018), DS6, 2018

[7] Harrary, Graph theory, Addison-Wesley, Reading, Massachusetts (1972).

[8] Jeena. R. V, S. Asha, Prime Mean Cordial Labeling of Graphs, European Chemical Bulletin, vol.12, pp.109-115, 2023.

[9] Raja Ponraj, Muthirulan SivaKumar, Murugesan Sundaram, Mean Cordial Labeling of Graphs, Open Journal of Discrete Mathematics,vol.2, No 145-148, 2012.

[10] Rosa, On certain valuations of the vertices of a graph, Theory of Graphs (International Symposium, Rome, July 1966), Gordon and Breach, N. Y. and Dunod Paris (1967) 349-355.