Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 1 (2024)

A Study on Emergence of Clutch Graphs from Cycle Graphs: A Comprehensive Analysis

Tamilselvi V[1], Thamizhendhi G[2]
[1] Assistant Professor of Mathematics, Vellalar College for Women(Autonomous), Erode, Tamilnadu, India,
Email id: tmlselvi18@gmail.com
[2] Assistant Professor of Mathematics, Sri Vasavi College, Erode, Tamilnadu, India,
Email id: gkthamil@gmail.com

Article History:

Received: *28-10-2023*

Revised: *09-12-2023*

Accepted: *24-12-2023*

Abstract:

The clutch is one of the substantial devices for constructing vehicles in automobile engineering. In terms of network toughness against failures or disruptions, the clutch-based graph can be used to model insufficiency of primary pathways and the availability of alternative routes in communication networks. In this paper, the identification of the clutch graph $Cl_{3n}(G)$ from the cycle graph $C_n(G)$ has been proposed. A clutch graph $Cl_{3n}(G)$ generating from a cycle graph with $3n$ vertices and $4n$ edges ($n \geq 4$ and even). The notions of degree, girth, and chromatic number of the clutch graph have been discussed. Further, the existence of the bipartite and Hamiltonian graphs on the clutch graph has been examined.

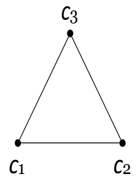
Keywords: Clutch graph, Cycle graph, Spanning tree, Bipartite, Hamiltonian.

1. Introduction

Graph theory explores connections between vertices and edges, offering a way to under- stand and analyze various real-world systems [6]. A cycle in graph theory is a closed path that starts and ends at the same vertex. A cycle graph is a graph composed of a single cycle, forming a circular structure [7]. A clutch is a mechanical device that connects or dis- connects power transmission, enabling smooth control over energy transfer [8]. In this way, the authors motivated to define clutch based graph that is used to design networks. Clutch graph based network models effective in identifying the smooth control of data transmission between entities. The clutch graph inherits the properties of bipartite and Hamiltonian graphs. The chromatic number and girth have been explored with appropriate illustration.

2. Preliminaries

Definition 2.1:[1] The *cycle* C_n , $n \geq 3$, made up of n vertices $c_1, c_2, \ldots, c_n$ and edges $\{c_1, c_2\},\{c_2, c_3\}, \ldots,\{c_{n-1}, c_n\},\{c_n, c_1\}$. The cycles c_3 and c_4 are shown in Figure 1 and Figure 2.



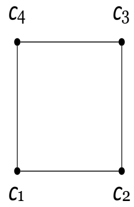


Figure 1: $C_3(G)$ Figure 2: $C_4(G)$

Definition 2.2:[1] A graph with a single cycle (at least 3 vertices) connected in a closed chain is referred to as a *cycle graph* C_n or circular graph. In C_n, the number of edges is equal to the number of vertices, and each vertex has degree 2.
Definition 2.3:[2] The number of edges that are incident on a vertex is the *degree* of the graph.
Definition 2.4:[5] The length of shortest cycle in the graph is said to be *girth*.
Definition 2.5:[1] The least number of colors that allows to be colored differently for the adjacent vertices is said to be *chromatic number*.

3. Main Results

Definition 3.1: A *Clutch Graph* $Cl_{3n}(G) = (V, E)$ is a type of graph that can be derived from the cycle graph. The construction of a clutch graph $Cl_{3n}(G)$ involves specific steps as described below.

- Start with a cycle graph $C_n(G) = (V_c, E_c)$, with vertex set $V_c = \{c_1, c_2, \ldots, c_n\}$,where ($n \geq 4$, even) is the number of vertices and edge set, $E_c = \{(c_i, c_{i+1}) | i \in \{1, 2, \ldots, n-1\}\} \cup (c_n, c_1)$
- Add a set of vertices, $V_p = \{p_1, p_2, \ldots, p_n\}$ to the existing vertex set V_c
- Establish the edge set, E_{cp} by joining each vertex c_i with its corresponding p_i, $E_{cp} = \{(c_i, p_i) | i \in \{1, 2, \ldots, n\}\}$
- Form an edge set, $E_p = \{(p_i, p_{i+1}) | i \in \{1, 3, 5, \ldots, n-1\}\}$
- Create another set of vertices, $V_q = \{q_1, q_2, \ldots, q_n\}$ and add to the vertex set V_p. The resulting vertex set be $V_{cpq} = \{c_1, c_2, \ldots, c_n, p_1, p_2, \ldots, p_n, q_1, q_2, \ldots, q_n\}$
- Built an edge set $E_{pq} = \{(p_i, q_i) | i \in \{1, 2, \ldots, n\}\}$ connecting corresponding vertices from V_p and V_q
- Finally, create an edge set, $E_q = \{(q_i, q_{i+1}) | i \in \{2, 4, \ldots, n\}\} \cup (q_n, q_1)\}$

By following these steps, the clutch graph $Cl_{3n}(G)$ have the vertex set $V = V_c \cup V_p \cup V_q$ and edge set $E = E_c \cup E_{cp} \cup E_p \cup E_{pq} \cup E_q$. The resulting clutch graph is characterized by with $|V| = 3n$ vertices and $|E| = 4n$ edges ($n \geq 4$ & *even*).
Example: The clutch graph $Cl_{18}(G)$ in Fig: 3 and $Cl_{24}(G)$ in Fig: 4 which have been constructed from the cycle graph $C_6(G)$ and $C_8(G)$ respectively.

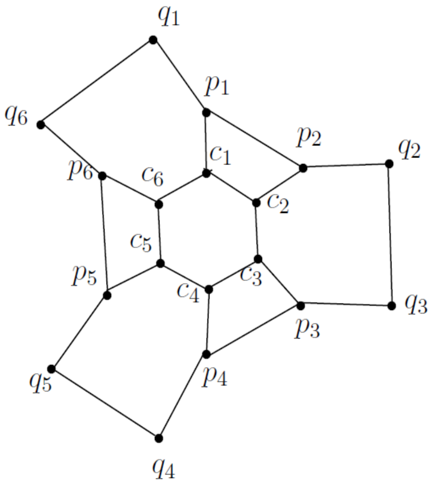


Figure 3: $Cl_{18}(G)$

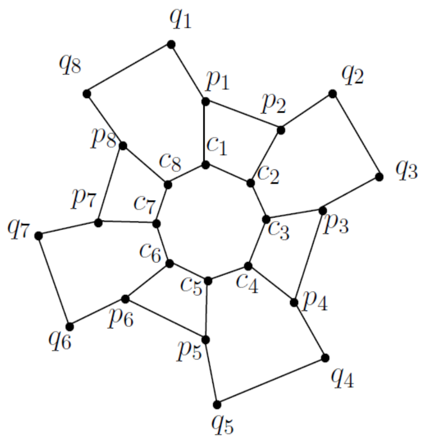


Figure 4: $Cl_{24}(G)$

Definition 3.2: The *degree* of the clutch graph is represented by *deg* $Cl_{3n}(G)$. The degree of the vertices c_i in V_c and p_i in V_p of the clutch graph are denoted by s. The degree of the vertices q_i in V_q is denoted by t.

$$deg_{c_i \in V_c} Cl_{3n}(c_i) = s = 3$$
$$deg_{p_i \in V_p} Cl_{3n}(p_i) = s = 3, \text{ and}$$
$$deg_{q_i \in V_q} Cl_{3n}(q_i) = t = 2$$

Definition 3.3: The *girth g* of a clutch graph ($Cl_{3n}(G)$) is always 4.

$$g = \{(c_{2i-i}, p_{2i-1}, p_{2i}, c_{2i}) | i \in 1, 2, \ldots, \tfrac{n}{2}\}$$

Definition 3.4: The *chromatic number* for clutch graph $\chi(Cl_{3n}(G))$ is 2.

$$\chi(Cl_{3n}(G)) = \begin{cases} 1 & if\ \{(c_{2i-1}, p_{2i}, q_{2i-1}) | i \in \{1, 2, \ldots, \frac{n}{2}\} \\ 2 & if\ \{(c_{2i}, p_{2i-i}, q_{2i}) | i \in \{1, 2, \ldots, \frac{n}{2}\} \end{cases}$$

Here 1,2 represents the colors assign to the vertices of the clutch graph.

4. Some properties of clutch graph

Theorem 4.1

For every clutch graph, the sum of the degrees of vertices is equal to twice the number of edges.

Proof

Let us consider a clutch graph $Cl_{3n}(G)$ with vertex set V_{cpq} where $V_{cpq} = (V_c \cup V_P \cup V_q)$. Each vertex c_i in V_c and p_i in V_P have degrees 3 and each vertex q_i in V_q have degrees 2.

$$\sum_{c_i \in V_c} deg(c_i) + \sum_{p_i \in V_p} deg(p_i) + \sum_{q_i \in V_q} deg(q_i) = 3n + 3n + 2n = 8n = 2(4n) = 2e.$$

Example: In Figure 3, the clutch graph Cl18 (G) with n = 6 vertices.

$$\sum_{c_i \in V_c} deg(c_i) + \sum_{p_i \in V_p} deg(p_i) + \sum_{q_i \in V_q} deg(q_i) == 18 + 18 + 12 = 48$$
$$= 2(24) = 2e.$$

Theorem 4.2

Let $Cl_{3n}(G)$ be a clutch graph with vertices of degrees s (or) t, then show that $n(2s+t) = 2 \in$, n is the number of vertices in the cycle graph.

Proof

Given that $Cl_{3n}(G)$ is a clutch graph whose vertices have degree s or t. The vertices in V_c and V_P have degrees s and V_q have degrees t respectively. By Theorem [4.1], given that,

$$\sum_{c_i \in V_c} \deg(c_i) + \sum_{p_i \in V_p} \deg(p_i) + \sum_{q_i \in V_q} \deg(q_i) = 2e$$
$$ns + ns + nt = 2 \in$$
$$n(2s + t) = 2 \in$$

Hence proved

Theorem 4.3

Every clutch graph $Cl_{3n}(G)$ is constructed with $2n$ vertices having of odd degree in V_c and V_p, and n vertices of even degree in V_q, results in a connected graph.

Proof

Given that $s = 3$ be the degree of each vertex c_i in V_c and each vertex p_i in V_p. Then total sum of odd degrees for these $2n$ vertices is

$$\sum_{i=1}^{2n} \deg(c_i) + \sum_{i=1}^{2n} \deg(p_i) = 2n.s = 6n$$

Also $t = 2$ be the degree of each vertex q_i in V_q. The total sum of even degrees for these n vertices is

$$\sum_{i=1}^{n} \deg(q_i) = n.t = 2n.$$

The total sum of degrees for all vertices is $6n + 2n = 8n$. According to the Handshaking Lemma, in a graph

$$\sum_{v \epsilon V} \deg CI_{3n}(G) = 2.|E| = 2.4n$$
$$\sum_{v \in V} \deg CI_{3n}(G) = 8n$$

The fact that $\sum_{v \in V} \deg CI_{3n}(G) = 8n$ implies $|E| = \frac{1}{2}\sum_{v \in V} \deg CI_{3n}(G) = 4n$ edges. This ensures that every vertex is incident to at least one edge, establishing connectivity in the clutch graph.

Theorem 4.4

For every clutch graph, $\chi + g \leq n + 2$, where χ is chromatic number, g is girth and n is number of vertices.

Example: In Fig: 3, the clutch graph $Cl_{18}(G)$ with n =6 vertices, resulting in $3n=18$ vertices. For every clutch graph, the chromatic number χ is 2, and the girth g is 4. Therefore,

$$\chi + g = 2 + 4 = 6$$
$$n + 2 = 6 + 2 = 8$$

The example satisfies the inequality $\chi + g \leq n + 2$, illustrating the validity of the theorem for this clutch graph with $n = 6$.

Theorem 4.5

In clutch graph $Cl_{3n}(G)$, the number of vertices with odd degree is even.

Proof

The clutch graph $Cl_{3n}(G)$ has $3n$ vertices. Since, the sum of degrees of all vertices in the clutch graph is $2e$, e is the number of edges.

$$\sum_{i=1}^{n} \deg(v_i) = 2e,\ v_i \in V = V_c \cup V_P \cup V_q$$

$$\sum_{c_i \in V_c} deg(c_i) + \sum_{p_i \in V_p} deg(p_i) + \sum_{q_i \in V_q} deg(q_i) = 2e$$

In the clutch graph, each vertex in V_c and V_p with degree 3, and each vertex in V_q with degree 2. Let n be the number of vertices of the vertex set V_c , V_p and V_q respectively.

$$3n + 3n + 2n = 2e$$

$$8n = 2e$$

Here the sum of degrees is even, and the degrees in V_q are all even, then the sum of degrees in V_c and V_p (which are all odd) must be even. It follows that, the number of vertices with odd degree is even.

Theorem 4.6

Every clutch graph $Cl_{3n}(G)$ has $78[4n-k]^{k+1}$ spanning trees.

Proof

Step 1: Consider the clutch graph $Cl_{3n}(G)$ with the vertex set $V = V_c \cup V_p \cup V_q$ and edge sets $E = E_c \cup E_cp \cup E_p \cup E_{pq} \cup E_q$.

Step 2: The degrees of the vertices in V are $deg(v_i) = \begin{cases} 3, & if\ v_i \in V_c \cup V_p \\ 2, & if\ v_i \in V_q \end{cases}$

Then, $D(Cl_{3n}(G))$ is the diagonal matrix of vertex degrees which is given by

$$D(Cl_{3n}(G)) = \begin{bmatrix} D_{cc} & 0 & 0 \\ 0 & D_{pp} & 0 \\ 0 & 0 & D_{qq} \end{bmatrix}$$

Step 3: Form an adjacency matrix based on the edges of the clutch graph

$$A(Cl_{3n}(G)) = \begin{cases} 1, & \text{if there is an edge between vertices } i \text{ and } j \\ 0, & \text{otherwise} \end{cases}$$

$$A(Cl_{3n}(G)) = \begin{bmatrix} A_{cc} & A_{cp} & 0 \\ A_{cp} & A_{pp} & A_{pq} \\ 0 & A_{Pq} & A_{qq} \end{bmatrix}$$

Step 4: The Laplacian matrix is defined as $L(Cl_{3n}(G)) = D(Cl_{3n}(G)) - A(Cl_{3n}(G))$. So

$$L(Cl_{3n}(G)) = \begin{bmatrix} D_{cc} - A_{cc} & -A_{cp} & 0 \\ -A_{cp} & D_{pp} - A_{pp} & -A_{pq} \\ 0 & -A_{Pq} & D_{qq} - A_{qq} \end{bmatrix}$$

Then, by Kirchoff's theorem, the number of distinct spanning tree of the graph is equal to any cofactors of its Laplacian matrix. Hence, the cofactor of $L(Cl_{3n}(G))$ is calculated by using $(-1)^{i+j} \cdot \det(L)$ that remains after deleting i^{th} row and j^{th} column. Therefore, the clutch graph has n distinct spanning trees.

Example: Let's find the cofactors of Laplacian matrix for the clutch graph $Cl_{12}(G)$ using Jupyter notebook software. Start with the construction of $Cl_{12}(G)$.

```
import matplotlib.pyplot as plt
import numpy as np

#Points
points={
    'c1' : (1.48, 0.41),
    'c2' : (2.44, 0.41),
    'c3' : (2.46, -0.41),
    'c4' : (1.46, -0.43),
    'p1' : (0.88, 0.79),
    'q2' : (3.46, 1.39),
    'p3' : (2.86, -0.81),
    'p4' : (0.84, -0.79),
    'q1' : (0.08, 1.37),
    'q3' : (3.46, 1.37),
    'p2' : (2.88, 0.81),
    'q4' : (0.06, -1.39)

  }

  # Plot points
  for point, coordinates in points.items():
  plt.scatter(*coordinates, label=point)
  # Connect points with lines
  lines = [
```

Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 1 (2024)

```
        (‘c1’,’c2’,’c3’,’c4’,’c1’),
        (’c1’, ’p1’),
        (’c2’, ’p2’),
        (’c3’, ’p3’),
        (’c4’, ’p4’),
        (’p1’, ’p2’),
        (’p3’, ’p4’),
        (’p1’, ’q1’),
        (’p2’, ’q2’),
        (’p3’, ’q3’),
        (’p4’, ’q4’),
        (’q2’, ’q3’),
        (’q4’, ’q1’)
]
for line in lines:
      plt.plot(*zip(*[points[point] for point in line]))

# Label points
for point, coordinates in points.items():
plt.annotate(point, coordinates,textcoords="offset points",xytext=(0, 5), ha=’center’)
# Add title below the graph
fig.text(0.5, 0.02, ’Graph Diagram’, ha=’center’, fontsize=12)

plt.legend()
plt.grid(True)
plt.show()
```

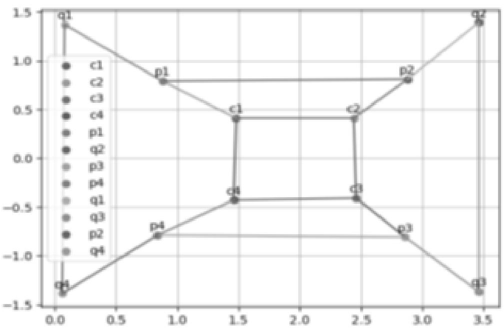


Figure 5: Cl_{12} (G)

First create a Diagonal matrix $D(Cl_{3n}(G))$.

```
import networkx as nx
import numpy as np

# Create a graph
G = nx.Graph()
edges = [(1, 2), (2, 3), (3, 4), (4, 1), (1, 5), (2, 6), (3, 7), (4, 8), (5, 6), (7, 8), (5, 9),
```

```
(6, 10), (7, 11), (8, 12), (10, 11), (12, 9)]
G.add_edges_from(edges)

# Get the nodes from the edges
nodes = set(node for edge in edges for node in edge)

# Create a diagonal matrix with zeros
diagonal_matrix = np.zeros((len(nodes), len(nodes)))

# Assign diagonal values
for i, node in enumerate(nodes):
diagonal_matrix[i, i] = G.degree(node)
# Assign a name to the matrix
D_Cl3n = diagonal_matrix

# Print the diagonal matrix with the assigned name
print("D(Cl_3n):")
print(D_Cl3n)
```

which display the output as

$$D(Cl_{3n}(G)) = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Create an Adjacency matrix $A(Cl_{3n}(G))$ using

```
import networkx as nx
import numpy as np

# Create a graph
G = nx.Graph()
G.add edges from([(1, 2), (2, 3), (3, 4), (4, 1), (1, 5), (2, 6), (3, 7), (4, 8),
(5, 6), (7, 8), (5, 9), (6, 10), (7, 11), (8, 12), (10, 11), (12, 9)])\\

# Obtain the adjacency matrix
adjacency matrix = nx.adjacency matrix(G).todense()

# Assign a name to the matrix
```

Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 1 (2024)

```
A(Cl_{3n}(G)) = $adjacency matrix $

# Print the adjacency matrix with the assigned name
print(f"A(Cl_{3n}(G))):\n{A(Cl_{3n}(G))"}
```

results in

$$A(\mathrm{Cl}_{3n}(G)) = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Next, the Laplacian matrix is given by $L(Cl_{3n}(G)) = D(Cl_{3n}(G)) - A(Cl_{3n}(G))$.

```
import networkx as nx
import numpy as np

# Create a graph
G = nx.Graph()
edges = [(1, 2), (2, 3), (3, 4), (4, 1), (1, 5), (2, 6), (3, 7), (4, 8), (5, 6),
(7, 8), (5, 9), (6, 10), (7, 11), (8, 12), (10, 11), (12, 9)]
G.add_edges_from(edges)

# Get the nodes from the edges
nodes = set(node for edge in edges for node in edge)


# Create the adjacency matrix
A_Cl3n = nx.adjacency_matrix(G).todense()

# Create the diagonal matrix
D_Cl3n = np.diag([G.degree(node) for node in nodes])

# Calculate the Laplacian matrix
 L_Cl3n = D_Cl3n - A_Cl3n

# Print the Laplacian matrix
print("L(Cl_3n):")
print(L_Cl3n)
```

$$L(\mathrm{Cl}_{3n}) = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 3 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 2 \end{bmatrix}$$

Now to find the cofactor of Laplacian matrix

```
import numpy as np

# Define the Laplacian Matrix L
L = np.array([
    [3, -1, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0],
    [-1, 3, -1, 0, 0, -1, 0, 0, 0, 0, 0, 0],
    [0, -1, 3, -1, 0, 0, -1, 0, 0, 0, 0, 0],
    [-1, 0, -1, 3, 0, 0, 0, -1, 0, 0, 0, 0],
    [-1, 0, 0, 0, 3, -1, 0, 0, -1, 0, 0, 0],
    [0, -1, 0, 0, -1, 3, 0, 0, 0, -1, 0, 0],
    [0, 0, -1, 0, 0, 0, 3, -1, 0, 0, -1, 0],
    [0, 0, 0, -1, 0, 0, -1, 3, 0, 0, 0, -1],
    [0, 0, 0, 0, -1, 0, 0, 0, 2, 0, 0, -1],
    [0, 0, 0, 0, 0, -1, 0, 0, 0, 2, -1, 0],
    [0, 0, 0, 0, 0, 0, -1, 0, 0, -1, 2, 0],
    [0, 0, 0, 0, 0, 0, 0, -1, -1, 0, 0, 2]
])

# Function to find the cofactor of a matrix
def cofactor(matrix, row, col):
minor_matrix = np.delete(np.delete(matrix, row, axis=0), col, axis=1)
sign = (-1) ** (row + col)
return sign * np.linalg.det(minor_matrix)

# Calculate the cofactor C12
row_index = 0
col_index = 1
cofactor_C12 = cofactor(L, row_index, col_index)
```

```
print(f"Cofactor C12: {cofactor_C12}")
```

Finally get the output C_{12}: 1248, which is one of the cofactor of the Laplacian matrix. Hence, for n = 4 in the clutch graph $Cl_{3n}(G)$, the number of spanning trees is 1248. Following the same approach, we found 41262 spanning trees for n = 6 and 210240 for n = 8. Based on these results, we derived a general formula for calculating the number of spanning trees in the clutch graph $Cl_{3n}(G)$ as $78[4n - k]^{k+1}$, where $k \in \{0, 1, 2, \ldots\}$. Therefore, the total number of spanning trees in the clutch graph $Cl_{3n}(G)$ can be expressed by the formula $78[4n - k]^{k+1}$.

Theorem 4.7

Every clutch graph $Cl_{3n}(G)$ is bipartite.

Proof

The clutch graph has three sets of vertices: V_c (cycle vertices), V_p (vertices introduced to the cycle), and V_q (vertices introduced to V_p). Now, consider two disjoint sets V_1 and V_2 such that $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$. Partitioned the vertex sets V_c, V_p, V_q in $Cl_{3n}(G)$ such that, each edge (v_i, v_j) in the clutch graph $Cl_{3n}(G)$ is of the form

$$\text{If } v_i \in V_1, \text{ then } v_j \text{ must be in } V_2$$
$$\text{If } v_j \in V_2, \text{ then } v_i \text{ must be in } V_1$$

The partition of the vertex set is

$$V_1 = \{(c_{2i-1}, p_{2i}, q_{2i-1}) \mid i \in \{1, 2, \ldots, \tfrac{n}{2}\}\}$$
$$V_2 = \{(c_{2i}, p_{2i-1}, q_{2i}) \mid i \in \{1, 2, \ldots, \tfrac{n}{2}\}\}$$

The condition holds for every edge in the clutch graph. The graph is bipartite iff it has no odd cycles. Based on the results, the clutch graph has no odd cycles. So, it is proved that every clutch graph is bipartite.

Theorem 4.8

Every clutch graph $Cl_{3n}(G)$, ($n \geq 4$, even) is Hamiltonian.

Proof

Consider the clutch graph $Cl_{3n}(G)$ with vertex set V and edge set E. Let us decompose the clutch graph into two components as G_1 representing the cycle graph $C_n(G)$ and G_2 representing the additional edges connecting V_p and V_q.

$$G_1 = (V_c, E_c)$$

$$G_2 = (V_p \cup V_q, E_{cp} \cup E_{pq} \cup E_q)$$

Since G_1 is a cycle graph, there exists a Hamiltonian cycle H_1. Further, Connecting the edges V_p and V_q form a Hamiltonian path P_2.

$$H_1 = (c_{i1}, c_{i2}, \ldots, c_{in}, c_{i1})$$

$$P_2 = (p_{j1}, q_{j1}, p_{j2}, q_{j2}, \ldots, p_{jn}, q_{jn})$$

Combine H_1 and P_2 to form a Hamiltonian cycle H for Cl_{3n} (G).

$$H = (c_{i1}, c_{i2}, ..., c_{in}, c_{i1}, p_{j1}, q_{j1}, p_{j2}, q_{j2}, ..., p_{jn}, q_{jn}).$$

As it concluded that every clutch graph with n vertices (n ≥ 4, even) has a Hamiltonian cycle.

Let's consider the case where n = 6 and construct the clutch graph Cl_{18} (G). The cycle graph with n = 6 has vertices $c_1, c_2, c_3, c_4, c_5, c_6$ and edges:

$$C_n = (c_1, c_2), (c_2, c_3), (c_3, c_4), (c_4, c_5), (c_5, c_6), (c_6, c_1)$$

- Add vertices $V_c = \{c_1, c_2, c_3, c_4, c_5, c_6\}$, $V_p = \{p_1, p_2, p_3, p_4, p_5, p_6\}$, and $Vq = \{q_1, q_2, q_3, q_4, q_5, q_6\}$.
- Connect each c_i to its corresponding pi: $E_{cp} = \{(c_1, p_1), (c_2, p_2), . . . , (c_6, p_6)\}$.
- Connect p_i to q_i: $E_{pq} = \{(p_1, q_1), (p_2, q_2), . . . , (p_6, q_6)\}$.
- Connect q_i to q_{i+1} (with q_6 connecting to q_1): $E_q = \{(q_2, q_3), (q_4, q_5), (q_6, q_1)\}$.
- Connect p_i to p_{i+1} (with p_6 connecting to p_1): $E_p = \{(p_1, p_2), (p_3, p_4), (p_5, p_6)\}$.

A Hamiltonian cycle can be traversed as

$$c_1 \to p_1 \to q_1 \to c_2 \to p_2 \to q_2 \to c_3 \to p_3 \to q_3 \to c_4 \to p_4 \to q_4 \to c_5 \to p_5 \to q_5 \to c_6 \to p_6 \to q_6 \to c_1$$

Conclusion

This paper introduces the concept of clutch graphs from cycle graphs. The clutch graph properties, such as degree, girth, and chromatic number have been discussed. Theo- rems are presented to illustrate the sum of the degrees and number of distinct spanning trees in the clutch graph. Moreover, the paper establishes the existence of bipartite and hamiltonian in the clutch graph. Furthermore, the author plans to apply these concepts to network analysis.

References

[1] Vasudev.C, Graph Theory and Applications, New Age International (P)Ltd, (2009).
[2] Bondy. J. A and Murty. U. S.R, Graph Theory, Springer International Edition, (2008).
[3] Harary. F, Graph Theory, Addison-Wesley,Reading, Mass, (1969).
[4] Kamran Azhar, Sohail Zafar, Agha Kashif, Amer Aljaedi, Umar Albalawi, The Application of Fault-Tolerant Partition Resolvability in Cycle-Related Graphs, Applied Science (2022), 12(19), 9558.
[5] Jonathan L. Gross, Jay Yellen, Mark Anderson, Graph Theory and its Applications, CRC Press, Taylor Francis Group, Mass, (2019).
[6] Badwaik Jyoti S. Recent Advances in Graph Theory and its Applications, International journal of scientific research in science, engineering and technology, February, (2020), Special Issue A7 (533-538).
[7] Jayesh Kudase1, Priyanka Bane, A Brief Study of Graph Data Structure, International Journal of Advanced Research in Computer and Communication Engineering, 5(6), (2016), ISSN (Print) 2319-5940.
[8] Sagar Jay Bhoite, Design and Analysis of Single Plate Friction Clutch, International Journal

of Innovative Research in Technology, February (2022), 8(9), ISSN: 2349-6002.

[9] Abhishek Chowdhary, Anupam Kumar, Sanket Kumar Singh, Design and Analysis of an Electro-Magnetic Clutch, International Journal of Progressive Research in Science and Engineering, June (2020), 1(3), (89-95).

[10] Manuel Tentarelli, Stefano Cantelli,Silvio Sorrentino, Alessandro De Felice, A New Approach to the Study and Prevention of the Clutch Judder, The American Society of Mechanical Engineers, February (2023), 1(3).