Proceedings of the 3rd ITB Graduate School Conference ISSN: 2963-718X
Enhancing Creativity in Research Through Developing Innovative Capabilities
December 21, 2022

Simulation of the Properties of the Coprime Graph of the Dihedral Group with Prime Order

Verrel Rievaldo Wijaya* & Abdul Gazir Syarifudin

Institut Teknologi Bandung, Jalan Ganesa No. 10 Bandung 40132, Indonesia
*Email: arywijaya86@gmail.com

Abstract. The dihedral group of order $2n$ denoted by D_{2n} is the symmetry group of a regular n-polygon consisting of rotation and reflection elements and the composition of both elements. We then construct a coprime graph of the dihedral group with prime order. Many properties can be studied from this coprime graph including the shape, spectrum, tree numbers, chromatic numbers, etc. By utilizing the Python programming language, we can draw this graph and simulate the properties concerning this graph to get more insight on it.

Keywords: *coprime graph; dihedral group; spanning tree; python, network.*

1 Introduction

Since the first coprime graph of a group was defined by Ma et al in 2014 [1], research on these topics has continued to grow. Starting from looking at the properties of coprime graphs from finite groups to special groups such as integer groups modulo n, dihedral groups to generalized quaternion groups. In 2020, Rina et al [2] examined the form of coprime graphs in the integer group modulo n. In the same year Syarifudin et al examined the shape of the coprime graph in the dihedral group, then continued to look at other properties such as the degree of vertex, radius, diameter, girth, clique number and chromatic number [3, 4, 5]. In 2021, Nurhabibah et al examined some properties of the coprime graph of the generalized quaternion group [6]. In addition, the parameters of the coprime graph in an infinite group were discussed, likes the so-called freedoms number.

Based on a lot of interest in obtaining properties on coprime graphs, for this reason in this study a program will be made to simulate the properties on coprime graphs. To be more specific to dihedral graphs with prime order, some of the properties simulated in this study are graph shape, adjacency matrix, graph spectrum, tree number and chromatic number.

2 Main Theory and Results

First, we give some definition and known results regarding coprime graph of dihedral group. Here and next, we will concern ourselves especially with dihedral group of prime order.

Definition 2.1 (Dihedral Group) [3] Group G is called a dihedral group of order $2n,\ n \geq 3$ and $n \in \mathbb{N}$, if it is being generated by two elements $a, b \in G$ with the properties

$$G = \langle a, b | a^n = e, b^2 = e, bab^{-1} = a^{-1} \rangle$$

Dihedrap group with order $2n$ will be denoted by D_{2n}.

From the definition it is easy to see that $|D_{2n}| = 2n$ and D_{2n}can be written as the set $D_{2n} = \{e, a, a^2, a^3, \dots, a^{n-1}, b, ab, a^2b, a^3b, \dots, a^{n-1}b\}$.

Definition 2.2 (Order) [7] Let G is a group with identity e and $x \in G$, the order of x is the smallest natural number k such that $x^k = e$ and denoted $|x| = k$.

Theorem 2.1 (Order of Element in D_{2n}) Let D_{2n} be a dihedral group. Then the order of its element is showed as follows

$$|a^i b^j| = \begin{cases} \dfrac{n}{\gcd(i, n)}, & j = 0 \\ 2, & j = 1 \end{cases}$$

where $a^i b^j \neq e$ dan $0 \leq i < n$

Next, we give definition of some properties concerning the algebraic graph theory that will be shown in the simulation.

Definition 2.3 (Coprime Graph) [1] The coprime graph of a group G, denoted by Γ_G is a graph whose vertices are elements of G and two distinct vertices u and v are adjacent if and only if $(|a|, |b|) = 1$.

Definition 2.4 (Adjacency Matrix) [8] The adjacency matrix of Γ is the $n \times n$ matriks $A = A(\Gamma)$ whose entries a_{ij} are given by

$$a_{ij} = \begin{cases} 1, & if\ v_i\ is\ adjacent\ to\ v_j \\ 0, & otherwise \end{cases}$$

Definition 2.5 (Incidence Matrix) [8] The incidence matrix D of Γ, with respect to a given orientation, is the $n \times m$ matrix d_{ij} whose entries are

$$d_{ij} = f(x) = \begin{cases} +1, & if\ v_i\ the\ positif\ end\ of\ e_j \\ -1, & if\ v_i\ is\ the\ negatif\ end\ of\ e_j \\ 0, & otherwise \end{cases}$$

Definition 2.6 (Spectrum) [8] The spectrum of a graph Γ is the set of numbers which are eigenvalues of $A(\Gamma)$, together with their multiplicities. If the distinct eigenvalues of $A(\Gamma)$ are $\lambda_0 > \lambda_1 > \cdots > \lambda_{s-1}$, and their multiplicities are $m(\lambda_0), m(\lambda_1)$, then we shall write

$$Spec\ \Gamma = \begin{pmatrix} \lambda_0 & \lambda_1 & \dots & \lambda_{s-1} \\ m(\lambda_0) & m(\lambda_1) & \dots & m(\lambda_{s-1}) \end{pmatrix}$$

Definition 2.7 (Spanning Tree) [8] A spanning tree in Γ is a subgraph which has $n - 1$ edges and contains no cycles.

Definition 2.8 (**Tree Number**) [8] The number of spanning trees of a graph Γ is called a tree-number, denoted by $\kappa(\Gamma)$.

Theorem 2.2 [8] Every cofactor of Q is equal to the tree-number of Γ, that is

$$adj\ Q = \kappa(\Gamma)\ \mathbf{J}$$

with **Q** is a Laplacian matrix of Γ and **J** is a matrix with all its entry 1.

Theorem 2.3 [8] Let the complete multipartite graph $K_{a_1,a_2,\dots,a_m}$, where $a_1 + a_2 + \cdots + a_m = n$, then $\kappa\left(K_{a_1,a_2,\dots,a_m}\right) = n^{m-2}(n - a_1)^{a_1-1} \dots (n - a_m)^{a_m-1}$

Theorem 2.4 Coprime graph of dihedral group D_{2n}, n a prime number is a complete tripartite graph $K_{1,n-1,n}$. Furthermore, the tree number of this graph is

$$\kappa\left(K_{1,n-1,n}\right) = 2(n + 1)^{n-2}n^n$$

Proof.

The dihedral group with n a prime number can be partitioned into three partition that is $\{e\}$, $\{a, a^2, \dots, a^{n-1}\}$, and $\{b, ab, \dots, a^{n-1}b\}$. These three partitions can easily be seen to be coprime with each other. So, the coprime graph is a complete tripartite graph $K_{1,n-1,n}$. Furthermore, using formula from Theorem 2.3 we get the tree number of this graph is

$$\begin{aligned} \kappa\left(K_{1,n-1,n}\right) &= (2n)^{3-2}(2n - 1)^{1-1}\left(2n - (n - 1)\right)^{n-1-1}(2n - n)^{n-1} \\ &= 2n(n + 1)^{n-2}n^{n-1} \end{aligned}$$

Definition 2.8 (Chromatic Numbers) [8] A proper coloring of a graph G is a coloring of the vertices of G so that no two adjacent vertices receive the same color. The chromatic number $\chi(G)$ is the least integer k such that there is a proper coloring of G using k colors.

Theorem 2.5 (Chromatic Numbers of D_{2n} Graph) [5] Let D_{2n} dihedral group. If $n = p_1^{k_1} p_2^{k_2} \dots p_m^{k_m}$, where p_i are distinct odd prime number then the chromatic numbers of coprime graph of dihedral group is $(m+2)$-partite.

3 Program Simulation

3.1 Python and Packages

In this paper, we will use programming language called Python to help simulating some properties of graph. Python is one of the most popular object-oriented high-level programming languages today and is widely used as a tool for mathematical computing and data analysis [9]. Previously, Python has been used to generate all subgroups of a particular group, for example Z_n [10].

Python contains package that can help us to work with studying graph and its properties. Some of the packages that will be used in our program are the following.

- **numPy**: fundamental package for scientific computing in Python
- **math**: provides access to many mathematical functions
- **matplotlib**: comprehensive library for creating static, animated, and interactive visualizations in Python.
- **tabulate**: creating table from another data
- **networkx**: library for studying graphs and networks

We are running our simulation in computer with processor AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx, 2100 Mhz, 4 Core(s), 8 Logical Processor(s).

```
In [1]: import numpy as np
        import networkx as nx
        import matplotlib.pyplot as plt
        from math import *
        from tabulate import tabulate
```

3.2 Group D_{2n} and the Order of each Element

First, we simulate the code to generate the group D_{2n} and the order of each element as follows.

```
In [2]: # Initialization
print("============================================")
print("Generating all the proper subgroups of D_2n")
print("============================================")
n = int(input("Input n: "))
print("D_%d = " %(2*n), end="")

# Group D_2n
keys = []
for i in range(2*n):
    if i==0:
        keys.append('e')
    elif i<n:
        keys.append("a^" + str(i))
    else: keys.append("a^" + str(i-n) +"b")
print(keys, "\n")

list_orde = {}
print('Order of each element')
for index, i in enumerate(keys):
    if index == 0:
        orde = 1
    elif index < n:
        orde = int(n / gcd(index, n))
    else:
        orde = 2
    list_orde[i] = orde
    print('| {} | = {}'.format(i, orde))
```

Simulations with an example $n = 11$ will obtained the following results.

```
============================================
Generating all the proper subgroups of D_2n
============================================
Input n: 11
D_22 = ['e', 'a^1', 'a^2', 'a^3', 'a^4', 'a^5', 'a^6', 'a^7', 'a^8', 'a^9', 'a^10', 'a^0b', 'a^1b', 'a^2b', 'a^3b', 'a^4b', 'a^
5b', 'a^6b', 'a^7b', 'a^8b', 'a^9b', 'a^10b']

Order of each element
| e | = 1
| a^1 | = 11
| a^2 | = 11
| a^3 | = 11
| a^4 | = 11
| a^5 | = 11
| a^6 | = 11
| a^7 | = 11
| a^8 | = 11
| a^9 | = 11
| a^10 | = 11
| a^0b | = 2
| a^1b | = 2
| a^2b | = 2
| a^3b | = 2
| a^4b | = 2
| a^5b | = 2
| a^6b | = 2
| a^7b | = 2
| a^8b | = 2
| a^9b | = 2
| a^10b | = 2
```

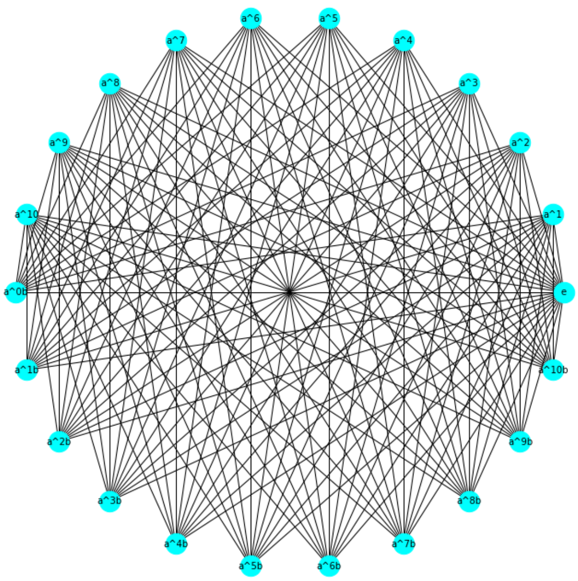
3.3 Drawing the Coprime Graph

First, we check whether the orders of the two elements in D_{2n} are coprime or not. If yes, then the two elements are connected by an edge in the graph. The following code checks and generates the edges of the coprime graph.

```
In [3]: all_edge = []

        for i in list_orde.keys():
            for j in list_orde.keys():
                if i != j:
                    if gcd(list_orde[i], list_orde[j]) == 1:
                        all_edge.append((i,j))
```

From the edge that has been obtained, the coprime graph can be drawn.

```
In [4]: G = nx.Graph()
        G.add_edges_from(all_edge)

        plt.figure(1,figsize=(10,10))
        nx.draw_circular(G, node_color = 'aqua',
                         with_labels=True, node_size=500, font_size=10)
        plt.show()
```



[redacted]Coprime graph of Dihedral Group D_{22}

3.4 Adjacency Matrix, Incidence Matrix, and Spectrum

Determining the adjacency matrix and incidence matrix of a graph can be done by utilizing the methods already available in the *networkx* package, namely *adjacency_matrix* and *incidence_matrix*.

```
In [5]: print('Adjacency matrix of G is ')
        print(nx.adjacency_matrix(G).toarray(), '\n')

        print('Incidence matrix of G is ')
        print(nx.incidence_matrix(G).toarray())
```

```
Adjacency matrix of G is
[[0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]]

Incidence matrix of G is
[[1. 1. 1. ... 0. 0. 0.]
 [1. 0. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 1. 0. 0.]
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 0. 0. 1.]]
```

The adjacency matrix of a graph is symmetric and real, so its eigenvalues are always real. The spectrum of this graph is as the following.

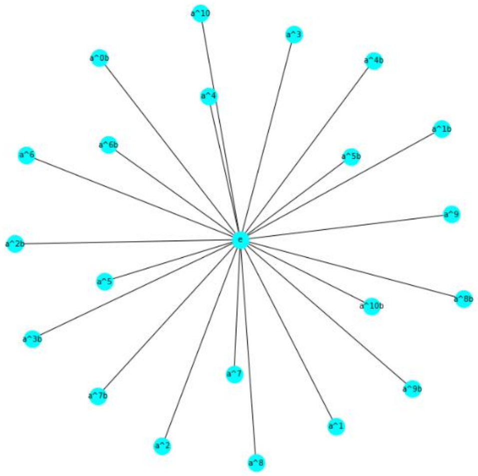
```
In [6]: print('Spectrum of G is ')

        unique_lambda = [np.real(i) for i in set(nx.adjacency_spectrum(G))]
        unique_lambda.sort(reverse=True)

        head = [['Lambda'], ['Count']]

        for index,i in enumerate(unique_lambda):
            head[0].append('{:.2e}'.format(np.real(i)))
            head[1].append(list(nx.adjacency_spectrum(G)).count(i))

        print(tabulate(head))
```

```
Spectrum of G is
------  ----  --------  --------  --------  --------  --------  -  ----------  --------  ---------  ---------  ---------  -----
-----
Lambda  12.2  1.94e-15  4.11e-30  3.59e-67  4.74e-81  3.01e-85  0  -5.71e-101  -1.9e-65  -6.79e-50  -7.25e-37  -1.52e-31  -1.72
-10.5
Count    1    1         1         1         1         1         9   1          1         1          1          1          1
1
------  ----  --------  --------  --------  --------  --------  -  ----------  --------  ---------  ---------  ---------  -----
```

3.5 Spanning Tree and Tree Number

Here we show one example of a spanning tree by utilizing the *minimum_spanning_tree* method in the *networkx* package as follows.

```
In [7]: T = nx.minimum_spanning_tree(G)
        plt.figure(1, figsize=(10,10))
        nx.draw(T, node_color = 'aqua', with_labels=True,
                node_size=500, font_size=10)
        plt.show()
```



[redacted]Spanning tree of Coprime graph

To find the number of all spanning tree of this graph we can use Theorem and Theorem from section 2. Using Theorem 2.2, we must find the determinant of cofactor of Laplacian matrix of the graph as follows.

```
In [8]: # Finding the degree matrix
        degree_m = np.zeros(shape=(2*n,2*n))

        index = 0
        for i in G.degree:
            degree_m[index, index] = i[1]
            index = index + 1
```

```
In [9]: laplacian_m = degree_m - nx.adjacency_matrix(G)
        laplacian_m_temp = laplacian_m[:, :2*n-1]
        laplacian_m_temp = laplacian_m_temp[:2*n-1,:]

        kappa1_G = int(np.linalg.det(laplacian_m_temp))
        kappa1_G

Out[9]: 2944291104429866221568
```

On the other hand, using the formula from Theorem 2.4 we get the results

```
In [10]: kappa2_G = 2* (n+1)**(n-2) * n**n
         kappa2_G

Out[10]: 2944291104429867270144
```

We also try to calculate this tree number for various value of prime n and we get the following result.

Table 1 The tree number of the Coprime graph

n	$\kappa_1(G)$	$\kappa_2(G)$
3	216	216
5	1350000	1350000
7	53971714048	53971714048
11	2944291104429866221568	2944291104429867270144
13	24530249648285413554855608 32	245302496482851820962602598 4

The actual real value of the tree number is the $\kappa_2(G)$. The results we get for $\kappa_1(G)$ is different because of the error in Python when calculating the determinant of big matrix. From this table, we can see that the tree number of coprime graph is increasing very fast.

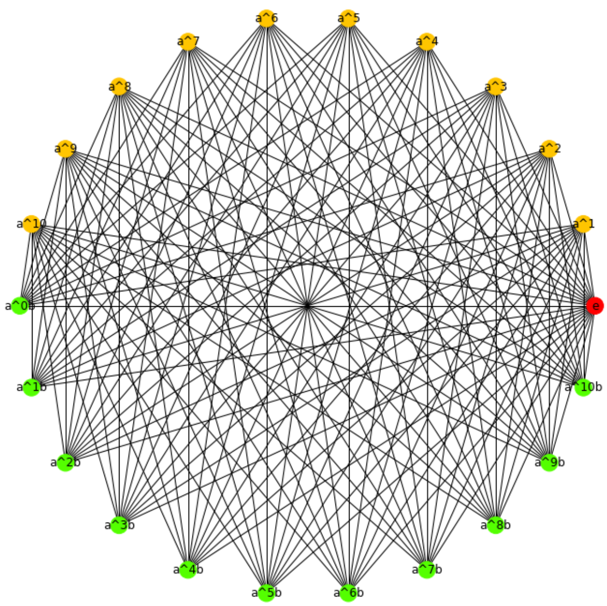
3.6 Chromatic Number

The chromatic number of a graph can be determined with the following code.

```
In [11]: H = nx.greedy_color(G)
         color = list(H.values())
         unique_color = set(color)
         print('The chromatic number of graph is', len(unique_color))

         The chromatic number of graph is 3
```

Based on the chromatic number, the code for coloring the graph so that the two neighboring vertex have different colors is as follows.

```
In [12]: def get_cmap(n, name='prism'):
             '''Returns a function that maps each index in 0, 1, ..., n-1 to a distinct
             RGB color; the keyword argument name must be a standard mpl colormap name.
             return plt.cm.get_cmap(name, n)

         cmap = get_cmap(len(unique_color))

         new_color = []
         for i in color:
             new_color.append(cmap(i))

         plt.figure(1, figsize=(10,10))
         nx.draw_circular(G, node_color = new_color, with_labels=True)
```



[redacted]Minimal coloring of the Coprime graph

4 Conclusion

Using the Python program, several properties of the coprime graph of the dihedral group can be seen, including the adjacency matrix, spectrum, tree number and chromatic number. Simulation using python helps in illustrating the graph image and minimal coloring of the graph. The tree number is also calculated with various prime n by two different formulas to compare its results. As seen in the table, the calculation of the tree number based on the determinant of the Laplacian cofactor matrix is still not accurate due to computational errors in Python.

References

[1] X. Ma, H. Wei & L. Yang, *The Coprime graph of a Group*, International Journal of Group Theory, **3**(3), pp. 13-23, 2014.

[2] R. Juliani, M. Masriani, I. G. A. W. Wardhana, N. W. Switrayni & I. Irwansyah, *Coprime Graph of Integers Modulo N Group and Its Subgroups*, Journal of Fundamental Mathematics and Applications (JFMA), **3**(1), pp. 15-18, 2020.

[3] A. G. Syarifudin, N. Nurhabibah, D. P. Malik & I. G. A. W. Wardhana, *Some characterizatsion of coprime graph of dihedral group D2n*, Journal of Physics: Conference Series, **1722**(1), 2021.

[4] A. G. Syarifudin, I. G. A. W. Wardhana, N. W. Switrayni and Q. Aini, *Some Properties of Coprime Graph of Dihedral Group D2n When n is prime power*, Journal of Physics: Conference Series, **1722**(1), 2021.

[5] A. G. Syarifudin, I. G. A. W. Wardhana, N. W. Switrayni and Q. Aini, *The Clique Numbers and Chromatic Numbers of the Coprime Graph of a Dihedral Group*, IOP Conference Series: Materials Science and Engineering, **1115**(1), 2021.

[6] N. Nurhabibah, A. G. Syarifudin and I. G. A. W. Wardhana, *Some Results of the Coprime Graph of a Generalized Quaternion Group Q_4n*, InPrime: Indonesian Journal of Pure and Applied Mathematics, **3**(1), pp. 29-33, 2021.

[7] J. A. Gallian, Contemporary Abstract Algebra 7th Edition, Belmont: Cengage Learning, 2009.

[8] N. Biggs, Algebraic Graph Theory, New York: Cambridge University Press, 1996.

[9] H. P. Langtangen, A Primer on Scientific Programming with Python 5th Edition, Springer Nature, 2016.

[10] I. B. Muktyas & S. Arifin, *Semua Subgrup Siklik dari Grup (Zn, +)*, Jurnal Teorema: Teori dan Riset Matematika, pp. 177-186, 2018.

[11] J. Hamm & A. Way, *Parameters of the Coprime Graph of a Group*, International Journal of Group Theory, **10**(3), 2021.