A Procedure for Determining The Locating Chromatic Number of An Origami Graphs

Agus Irawan[1*], Asmiati[2], Bernadhita Herindri Samodra Utami[1], Aang Nuryaman[2] and Kurnia Muludi[3]
[1]Information System, Institut Bakti Nusantara, PSDKU : Jl. Wismarini No. 09 Pringsewu, Lampung, Indonesia
[2]Department of Mathematics, Faculty of Mathematics and Natural Sciences, Lampung University, Jl. Brodjonegoro No.1 Bandar Lampung, Indonesia.
[3]Computer Sciences, Faculty of Mathematics and Natural Sciences, Lampung University, Jl. Brodjonegoro No.1 Bandar Lampung, Indonesia.
[*]Coreponding author : agusirawan814@gmail.com

Abstract
The concept of locating chromatic number of graph is a development of the concept of vertex coloring and partition dimension of graph. The locating-chromatic number of G, denoted by $\chi_L(G)$ is the smallest number such that G has a locating k-coloring. In this paper we will discussed about the procedure for determine the locating chromatic number of Origami graph using Python Programming.

Keywords:
locating-chromatic number, graph, origami, python.

1. Introduction

One of the studies in graph theory is the metric dimension which were first introduced by Slater and Harary [1] and Melter [2]. The application of metric dimensions plays a role in robotic navigation, optimization of the placement of threat detection sensors, and classification of chemical compound data [3]. Furthermore, in 2002 Chartrand et al. [4] for the first time studied the locating chromatic number of a graph with obtained two graph concepts, coloring vertices and partition dimension of a graph.

Chartrand et al. [4] give definition locating chromatic number of a graph. Suppose $G = (V, E)$ is a connected graph and c is a vertex coloring in G using the k colors of G. Suppose $\Pi = \{C_1, C_2, \dots, C_k\}$ is a partition of $V(G)$ which is induced by coloring c. The color code, $c_\Pi(v)$ of v is k-pairs ordered $(d(v, C_1), d(v, C_2) \dots d(v, C_k))$ with $d(v, C_i) = \min \{d(v, x) | x \in C_i\}$ for $1 \le i \le k$. If all vertices in $V(G)$ have different color codes, then c is called k-locating coloring of G. Next, They determined the locating chromatic number for path graphs, cycle graphs, and double star graphs. Furthermore, Chartrand et al. [5] characterized all graphs of order n with locating chromatic number $n - 1$.

The locating chromatic number of some Buckminsterfullerene-type graph introduced by Putri et al. [6]. Next, Rahimah et al. [7] obtained locating chromatic number for thorn graph of wheel graph W_3. On the results of certain operations, Asmiati et al. [8] found the locating chromatic number of certain barbell graphs. Furthermore, Asmiati et al. [9] obtained the locating chromatic number of subdivision of barbell graphs containing generalized Petersen graph.

The following definition of the locating chromatic number of a graph and origami graph [4, 10]. Furthermore, the locating chromatic number of Origami graphs was introduced by Irawan et al.[11]. Next, irawan et al. [12] found certain barbell Origami graphs and subdivision of certain barbell Origami graphs.

Theorem 1.1. [11] Let O_n be an Origami graph for $n \ge 3$. The locating chromatic number of an origami graphs O_n is 4 for $3 \le n \le 6$ and 5 otherwise.

The locating chromatic number of a graph is a newly interesting topic to study because there is no general theorem for determining the locating chromatic number of any graph. In this research, we specifying about the procedure for determine the locating chromatic number of Origami graph using Python Programming.

2. Methods

The steps taken to determine locating chromatic number of Origami graphs for some value of n is.

a. To constuct Origami graphs.

The following definition of Origami graph taken from[10]. An origami graph O_n, $n \ge 3$ is a graph with $V(O_n) = \{u_i, v_i, w_i : i \in \{1, \dots, n\}\}$ and $(O_n) = \{u_i w_i, u_i v_i, v_i w_i, u_i, u_{i+1}, w_i u_{i+1} : i \in \{1, \dots, n-1\} \cup \{u_n u_1, w_n u_1\}$.

b. Determine proposed algorithm

The algorithm was implemented in Python by following the results by Irawan et al. [11]. This program needs one input variable n that respectively represents a number in O_n. These inputs are then managed by the computer program to generate the Locating chromatic

Manuscript received September 5, 2022
Manuscript revised September 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.9.5

number of Origami graphs O_n.The main algorithm is given as follows.

```
main algoritma:

Get input of n of On from user
Draw the Origami graph from input user
Total_of_all_nodes_of_graph ← n*3
create array list_of_nodes from graph
On

// Do looping from n-1 until 4:
Repeat from a
←( Total_of_all_nodes_of_graph -1) to
4:
  next_locating-chromatic_number ←
false
  list_of_combinations ← {}
  Repeat from b ← 1 to a:
    Inspect The_total_colors for color
to-b
    Append the_total_colors in array
list_of_combinations
    Do repeat
  Repeat from b ← 1 to
length_of(list_of_combinations):
    Color_position ←
Permutation(list_of_combinations [b])
    Repeat from c ←1 to
length_of(color_position):
      set graph_color according to
color_position [c]
      invalid_neighbors ← false
      Repeat from d=1 to
length_of(node_list):
        Inspect neighbor_color for
node_list [d]
        If color node_list [d] =
neighbor_color node_list [d]:
          invalid_neighbors = true
          Break
        Do repeat
      If invalid_neighbors = false:
        array result ← {}
        Repeat from d=1 to
length_of(node_list):
          Inspect shortest_path from
node_list [d] to color_position [c]
          Append the result
shortest_path to array result
          Do repeat
        Find Duplicated from result
        If Not Duplicated:
          print result
          next_locating-
locating_chromatic_number ← true
          Break
      Do repeat
    If next_locating-chromatic_number =
true
      Break
    Do repeat
  If next_Locating_chromatic_number =
false
    String ← "The locating-chromatic
number is" + (a+1)
    print String
    Break
  Do repeat
End of program
```

c. Run the program to get an locating chromatic number of Origami graphs.

3. Results and Discussions

In this research, Python code is used to process the result using the hardware and software shown in Table 1.

Table 1 : System Specification

Items	Specification
Operating System	Debian 10
Processor	Intel® core™ i3 CPU 2.3 GHz.
Memory (RAM)	4 GB DDR3
OS Architecture	64 bit
Python	3.7.3 [GCC 8.3.0]
Libraries	ipython 7.13.0 jupyter 1.0.0 networkx 2.4 notebook 6.0.3

These results have been collected to compare the running time, number of vertex and the locating chromatic number for each type of Origami graph. In this work, the results are generated for the Origami graph of type O_3, O_4, and O_5. This restriction is applied due to the limitation of the system for running the code. The execution for a more complex type of the Origami graph will require more resources than such a system.

The summary of the execution time and the number of the locating chromatic number for each type of Origami graph is reported in Table 2. The O_3 has locating chromatic number is 4 that can be obtained in 17 second. When the vertex Origami graph is added on one vertex to O_4 the computation time is 43 second and locating chromatic number is 4 are successfully identified. A significant increase on execution time for Origami graph of O_5. In this

case, the execution time becomes 335 second, has locating chromatic number is 4. Each value of n Origami Graph added by one will increase execution time by an exponential rate in time. This makes sense since the code is running a brute force approach.

The following are the results obtained from several cases, namely O_3, O_4, and O_5, which are presented in table 2.

Table 2 : Measurement Results

Case	Type of Origami Graph	Number of Vertices	Execution Time	Locating-Chromatic Number
1	O_3	9	17 second	4
2	O_4	12	43 seconds	4
3	O_5	15	335 seconds	4

Next, the measurement results of O_3, O_4, and O_5, will be presented in the following curve :

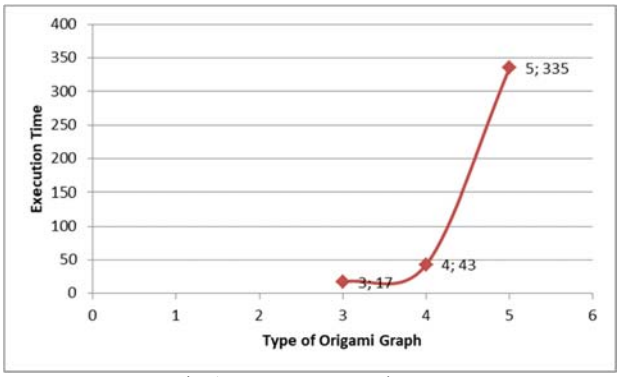

Fig. 1 Measurement result curve

Case 1 $\chi_L(O_3) = 4$
The Partition Π of $V(O_3)$:

$C_1 = \{u_1\}$;
$C_2 = \{u_3, v_2\}$;
$C_3 = \{u_2, v_1, v_3\}$;
$C_4 = \{w_1, w_2, w_3\}$.

Therefore the color codes of all vertices of O_3 are:
$c_\Pi(u_1) = (0,1,1,1)$; $c_\Pi(u_2) = (1,1,0,1)$; $c_\Pi(u_3) = (1,0,1,1)$; $c_\Pi(v_1) = (0,2,0,1)$; $c_\Pi(v_2) = (2,0,1,1)$; $c_\Pi(v_3) = (2,1,0,1)$; $c_\Pi(w_1) = (1,2,1,0)$; $c_\Pi(w_2) = (2,1,1,0)$; $c_\Pi(w_3) = (1,1,1,0)$.

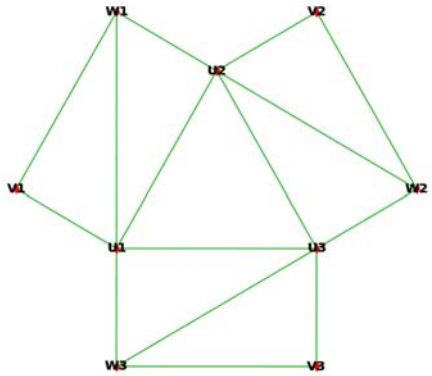

Fig. 2 An Origami graph O_3.

Case 2 $\chi_L(O_4) = 4$
The Partition Π of $V(O_4)$:

$C_1 = \{v_1, w_4\}$;
$C_2 = \{u_1, u_3, v_2\}$;
$C_3 = \{u_2, u_4, v_3\}$;
$C_4 = \{v_4, w_1, w_2, w_3\}$.

Therefore the color codes of all vertices of O_4 are:
$c_\Pi(u_1) = (1,0,1,1)$; $c_\Pi(u_2) = (2,1,0,1)$; $c_\Pi(u_3) = (2,0,1,1)$; $c_\Pi(u_4) = (1,1,0,1)$; $c_\Pi(v_1) = (0,2,0,1)$; $c_\Pi(v_2) = (3,0,1,1)$; $c_\Pi(v_3) = (3,1,0,1)$; $c_\Pi(v_4) = (1,2,0,1)$; $c_\Pi(w_1) = (1,1,1,0)$; $c_\Pi(w_2) = (3,1,1,0)$; $c_\Pi(w_3) = (2,1,1,0)$; $c_\Pi(w_4) = (0,1,1,1)$.

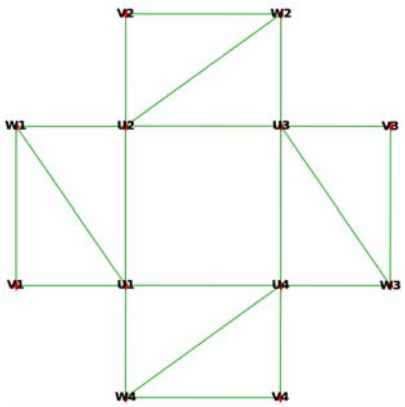

Fig. 3 An Origami graph O_4.

Case 3 $\chi_L(O_5) = 4$
The Partition Π of $V(O_3)$:

$C_1 = \{u_1, u_4, v_5\}$;
$C_2 = \{u_2, v_1, v_3\}$;
$C_3 = \{u_3, u_5, v_2, v_4\}$;
$C_4 = \{w_1, w_2, w_3, w_4, w_5\}$.

Therefore the color codes of all vertices of O_5 are: $c_\Pi(u_1) = (0,1,1,1)$; $c_\Pi(u_2) = (1,0,1,1)$; $c_\Pi(u_3) = (1,1,0,1)$; $c_\Pi(u_4) = (0,2,1,1)$; $c_\Pi(u_5) = (1,2,0,1)$; $c_\Pi(v_1) = (1,0,2,1)$; $c_\Pi(v_2) = (2,1,0,1)$; $c_\Pi(v_3) = (2,0,1,1)$; $c_\Pi(v_4) = (1,3,0,1)$; $c_\Pi(v_5) = (0,3,1,1)$; $c_\Pi(w_1) = (1,1,2,0)$; $c_\Pi(w_2) = (2,1,1,0)$; $c_\Pi(w_3) = (1,1,1,0)$; $c_\Pi(w_4) = (1,3,1,0)$; $c_\Pi(w_5) = (1,2,1,0)$.

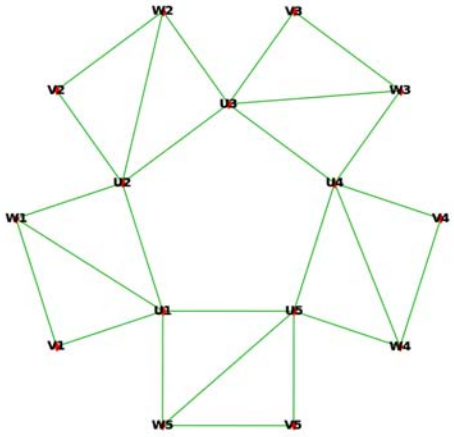

Fig. 4 An Origami graph O_5.

4. Conclusion

The locating chromatic number of an Origami graph can be defined by exact is 4. This study successfully implemented the locating chromatic number of O_3 to O_5 using Python code.

References

[1] F. Harary and R. Melter, On the metric dimention of a graph, Ars Combinatoria, Vol. 2, pp. 191-195, 1976.

[2] P. Slater, Leaves of trees, Congressus Numerantium, Vol. 14, pp. 549-559, 1975.

[3] V. Saenpholphat and P. Zhang, Conditional resolvability: a survey, International Journal of Mathematics and Mathematical Sciences, vol.38, pp.1997-2017, 2004.

[4] G. Chartrand, D. Erwin, M. A, Henning, P. J. Slater, P. Zhang, The locating-chromatic number of a graphs, Bulletin of the Institute of Combinatorics and its Applications, vol. 36, pp.89-101, 2002.

[5] G. Chartrand, D. Erwin, M. A, Henning,P. J. Slater, P. Zhang, Graf of order n with locating-chromatic number $n - 1$, Discrate Mathematics, vol.269, no.1-3, pp.65 – 79, 2003.

[6] Y S Putri, L Yulianti1 , and Yanita, On the locating chromatic number of some Buckminsterfullerene-type graphs, Journal of Physics: Conference Series, 1836, pp.1 – 7, 2021.

[7] E. Rahimah, L. Yulianti and D. Welyyanti, Determination the locating-chromatic number of thorn graph from a wheel graph W_3, Jurnal Matematika UNAND, vol.VII , pp.1 – 8, 2018.

[8] Asmiati, I. K. S. G. Yana, and L. Yulianti, On The Locating Chromatic Number of Certain Barbell Graphs, International Journal of Mathematics and Mathematical Sciences, pp.1-5, 2018.

[9] Asmiati, I. K. S. G. Yana, and L. Yulianti, On the locating chromatic number of subdivision of barbell graphs containing generalized petersen graph, International Journal of Computer Science and Network Security, Vol. 19 No. 7 pp. 45-50, 2019.

[10] S. Nabila, A. N. M.Salman, The Rainbow Conection Number of Origami Graphs and Pizza Graphs, Procedia Computer Science, 74, pp.162-167, 2015.

[11] A. Irawan, Asmiati, L. Zakaria, K. Muludi, The locating-chromatic number of origami graphs, Algorithms, vol.14, no.167, pp.1-15, 2021.

[12] A. Irawan, Asmiati, L. Zakaria, K. Muludi and B. H. S. Utami, Subdivision of Certain Barbell Operation of Origami Graphs has Locating-Chromatic Number Five, International Journal of Computer Science and Network Security, Vol. 21 No. 9 pp. 79-85, 2021.