J. Indones. Math. Soc.
Vol. 31, No. 04 (2025), pp. 1–9.

The Locating Chromatic Number For Corona Operation Of Path And Cycle With Python Programming

Nur Hamzah[1], Asmiati[2*], Aang Nuryaman[3]

[1,2,3]Department of Mathematics, Universitas Lampung, Indonesia
[1]hamzahnur6@gmail.com, [2]asmiati.1976@fmipa.unila.ac.id,
[3]aang.nuryaman@fmipa.unila.ac.id

Abstract. Locating chromatic number of a graph is a concept that is still interesting today because there is no theorem or algorithm that can determine the locating chromatic number of any graph. In this research, an algorithm was created to determine the locating chromatic number for corona operation of path and cycle with Python programming.

Key words and phrases: locating-chromatic number, corona operation, path, cycle, python.

1. INTRODUCTION

Given $G = (V, E)$ is a connected graph, where $V(G)$ is the nonempty set of elements called vertices and $E(G)$ is the set of unordered pairs $(u,\ v)$, which are called edges with $u,\ v \in V(G)$. Chartand, et al.[1] in 2002 introduced the locating chromatic number of a graph G, defined as follows.

Let S_i be a set of vertices that receive a color i and $\prod = \{S_1,\ S_2,\ S_3,\ \ldots\ , S_k\}$ is a set of color classes from $V(G)$. The color code for the vertex v denoted by $c_\Pi(v)$ is the k ordered pairs $(d\,(v, S_1)\,,\ d\,(v, S_2)\,,\ d\,(v, S_3)\,,\ \ldots,\ d(v, S_k)$ with $d\,(v, S_i) = \min\{d(v, x)|\, x \in S_i\}$ for $1 \leq i \leq k$ where $d(v,\ x)$ is the minimum distance from vertex v to vertex x. If each vertice of G has a different color code, then c is called the locating coloring of G. The minimum number of colors used in coloring of G is called the locating chromatic number (lcn) of G, denoted by $\chi_L(G)$.

Chartrand et al. [1] obtained $3\ \leq \chi_L(G) \leq n$ for arbitrary connected graph G. Next, they determined the lcn of path, $\chi_L\ (P_n) = 3$ for $n \geq 3$; $\chi_L(C_n) = 3$ for odd n and 4 for otherwise, where C_n is a cycle graph with n vertices. Chartrand et

*Corresponding author
2020 Mathematics Subject Classification: 05C15, 05C78.
Received: 05-01-2025, accepted: 31-05-2025.

al. [2] characterized graphs having *lcn* $(n-1)$ and have the upper bound $(n-2)$ for graph G with n vertices.

Asmiati determined the *lcn* for non homogeneous caterpillar graphs [3], barbell shadow path graphs [4], and certain operations of origami graphs [5]. However, Amanah et al. [6] determined *lcn* for amalgamation of some complete graphs. In 2021, Irawan et al. [7] determined the *lcn* of origami graphs and [8] using Python programming.

Research on the locating chromatic number of graph operations is still interesting today. Damayanti et al. [9] analyzed modified paths with cycles having *lcn* 4 and Prawinasti et al. [10] for split graphs of cycles. Zikra et al. [11] investigated disjoint unions of fan graphs, examining how to generate the *lcn* of this graph. Salindeho et al. [12] studied the *lcn* of subdivisions of friendship graphs, Welyyanti et al. [13] for disconnected graphs with components consisting of paths and cycles, and Rahmatalia et al. [14] for split path graphs.

One of the operation of graphs is the corona operation introduced by Fucht and Harary in 1970[15]. The corona operation of P_n and C_m, denoted by $P_n \odot C_m$ is defined as the graph obtained by taking one copy of P_n and $|V(P_n)|$ copies of C_m and then joining all vertices of the k^{th} copy of C_m with the k^{th}-vertex of P_n.

In the previous research, we determined *lcn* for the corona operation of the path and cycle manually and got the following result [16].

Theorem 1.1. *The lcn of* $P_n \odot C_3$ *is* 5 *for* $3 \leq n < 7$ *and 6 for* $n \geq 7$.

Theorem 1.2. *The lcn of* $P_n \odot C_4$ *is* 5 *for* $3 \leq n < 6$ *and 6 for* $n \geq 6$.

In this research, we discuss and justify these problems with Python programming.

2. METHODS

The steps taken to determine *lcn* of $(P_n \odot C_m)$ for $m, n \geq 3$ are as follows.

(1) Generate a graph from the corona operation of the path and cycle.
From[16] we have
$V(P_n \odot C_m) = \{v_k;\ k \in [1,\ n]\} \cup \{u_k^l;\ k \in [1,\ n],\ l \in [1,\ m]\}$ and
$E(P_n \odot C_m) = \{v_k\ v_{k+1};\ k \in [1,\ n-1]\} \cup \{u_k^l\ u_k^{l+1};\ k \in [1,\ n],\ l \in [1,\ m-1]\} \cup \{u_k^1\ u_k^l;\ k \in [1,\ n],\ l = m\} \cup \{v_k\ u_k^l;\ k \in [1,\ n],\ l \in [1,\ m]\}$.

(2) Determine proposed algorithm:

```
begin program
Input n of Pn from user
Input m of Cm from user
Input initial value locating chromatic number from user
Graph ← Generate graph Pn corona Cm
Do lopping from 1 to length-of(color−combination):
  Do looping from 1 to length-of(color−permutation):
  Color−permutation ← permutation(initial−value)
```

```
    nodelist(Graph)
    for i in color-permutation:
     if invalid-neighboor(i) ← True
     repeat
     color-code ← shortes-path from nodelist to a in
     color—permutation
      if invalid-lcn(color-node) ← True
      repeat
    return (color-code, color-permutation)
  return (color-code, color-permutation, color-combination)
  lcn ← length-of(color-permutation) + 1
  print("locating chromatic number is", lcn)
  print("list of node color", color-permutation)
  print("color code is", color-code)
  end program
```

(3) Run the program to get *lcn* of corona operation of path and cycle.

3. MAIN RESULTS

This research uses a computer system with the following specifications:

TABLE 1. System Specification

Item	Specification
Operating System	Microsoft Windows 10 Pro
Processor	Intel(R) Core(TM) i5-2430M CPU @ 2.40GHz, 2401 Mhz, 2 Core(s)
Memory(RAM)	4.00 GB
OS Architecture	64 bit
Python Version	3.11.7
Libraries	jupyter 2
	networkx 2.4

The results of running this program are collected in the form of execution time, number of vertices, and *lcn* for each type of corona operation of path and cycle. In this work, the results are described and compared for the corona operation of path and cycle with $n = 2, 3, 4$ and $m = 3, 4$. This restriction is applied because of the limitation of the system that enables us to run the code. Execution for more complex graph types will require more resources than the system specification. A summary of the execution time and the number of location chromatic numbers for each type of graph is given in Table 2. Each value of n of the corona operation of path and cycle added by one will increase the execution time at an exponential rate over time.

TABLE 2. Measurement Results

Case	Type of Graph	Number of Vertices	Execution Time	lcn
1	$P_2 \odot C_3$	8	10 s	5
2	$P_3 \odot C_3$	12	1 m 58 s	5
3	$P_4 \odot C_3$	16	1 h 14 m 23 s	5
4	$P_2 \odot C_4$	10	1 m 44 s	5
5	$P_3 \odot C_4$	16	1 h 5 m 13 s	5

Table 2 shows the results of the measurement of execution time to find the lcn of various types of graph resulting from corona operations. The table explains the number of vertices, execution time, and lcn value for each case. In the first case with the graph $P_2 \odot C_3$, which has 8 vertices, the execution time is 10 seconds, with an lcn value of 5. In the second case with the graph $P_3 \odot C_3$, the number of vertices increases to 12, so the execution time increases to 1 minute 58 seconds, but the value of lcn remains 5. The third case, the graph $P_4 \odot C_3$, has 16 vertices, and its execution time increases drastically to 1 hour 14 minutes 23 seconds. Although the execution time increases significantly, the lcn value remains 5.

In the fourth case, the graph $P_2 \odot C_4$ has 10 vertices with an execution time of 1 minute 44 seconds and an lcn value of 5. Finally, for $P_3 \odot C_4$, the number of vertices is 16 with an execution time of 1 hour 5 minutes 13 seconds, and the value of lcn also remains 5. A significant increase in execution time is seen in cases with a larger number of vertices. This shows that the addition of vertices in the corona operation graph directly affects the computational complexity, although the lcn value remains constant for all cases. The graphs accompanying this table can help clarify the differences in execution time between these cases.

The graphic image below can make it easier for us to see the clear difference when additional vertices occur in the corona operation result graph.

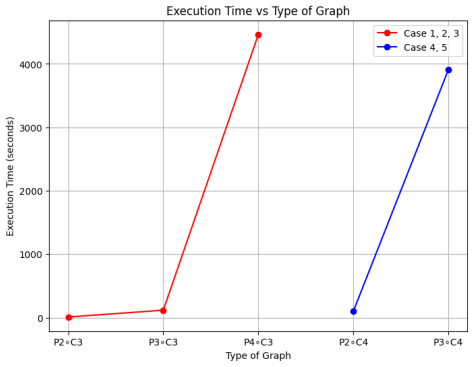

FIGURE 1. Measurement result curve of $P_n \odot C_3$ VS $P_n \odot C_4$

The curve on Figure 1 shows the relationship between the type of graph and the execution time in finding *lcn*. The horizontal axis represents the type of graph resulting from the corona operation, while the vertical axis shows the execution time in seconds.

This graph is divided into two groups on the basis of the type of graph. The red curve (cases 1, 2, and 3) represents the graph $P_n \odot C_3$. The blue curve (cases 4 and 5) represents the graph $P_n \odot C_4$.

In the red curve, there was a significant increase in execution time as the number of vertices increases in the graphs. For $P_2 \odot C_3$, the execution time is only 10 seconds, but increases dramatically to more than 4000 seconds (about 1 hour 14 minutes) at $P_4 \odot C_3$. In the blue curve, a similar pattern is also seen. The execution time for $P_2 \odot C_4$ is about 104 seconds (1 minute 44 seconds), and increases to more than 3900 seconds (about 1 hour 5 minutes) for $P_3 \odot C_4$.

This Figure 1 illustrates that the greater number of vertex in the graph of the corona operation causes a significant surge in execution time. This time increase is more clearly seen in the graph $P_n \odot C_3$ compared to $P_n \odot C_4$, although growth trends in general remain consistent for both groups.

The following are some results for determining the *lcn* of $P_n \odot C_m$ for $n = 2, 3, 4$ and $m = 3, 4$ using Python programming.

Case 1 $P_2 \odot C_3$

The partition Π of $V(P_2 \odot C_3)$:

$S_1 = \{v_2\}$
$S_2 = \{v_1\}$
$S_3 = \{u_3^1, u_3^2\}$
$S_4 = \{u_1^1, u_2^2\}$
$S_5 = \{u_2^1, u_1^2\}$.

Then, the color codes are:
$c_\Pi(u_1^1) = (2, 1, 1, 0, 1); c_\Pi(u_2^1) = (2, 1, 1, 1, 0); c_\Pi(u_3^1) = (2, 1, 0, 1, 1);$
$c_\Pi(u_1^2) = (1, 2, 1, 1, 0); c_\Pi(u_2^2) = (1, 2, 1, 0, 1); c_\Pi(u_3^2) = (1, 2, 0, 1, 1);$
$c_\Pi(v_1) = (1, 0, 1, 1, 1); c_\Pi(v_2) = (0, 1, 1, 1, 1).$

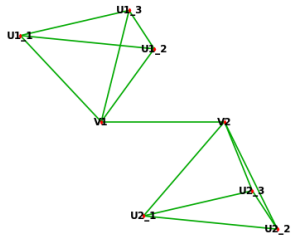

FIGURE 2. A minimum locating coloring of $P_2 \odot C_3$

Case 2 $P_3 \odot C_3$
The partition Π of $V(P_3 \odot C_3)$:
$S_1 = \{v_1, v_3\}$
$S_2 = \{u_3^3, v_2\}$
$S_3 = \{u_3^1, u_3^2\}$
$S_4 = \{u_1^1, u_2^2, u_2^3\}$
$S_5 = \{u_2^1, u_1^2, u_1^3\}$.
Then, the color codes are:
$c_\Pi(u_1^1) = (1, 2, 1, 0, 1); c_\Pi(u_2^1) = (1, 2, 1, 1, 0); c_\Pi(u_3^1) = (1, 2, 0, 1, 1);$
$c_\Pi(u_1^2) = (2, 1, 1, 1, 0); c_\Pi(u_2^2) = (2, 1, 1, 0, 1); c_\Pi(u_3^2) = (2, 1, 0, 1, 1);$
$c_\Pi(u_1^3) = (1, 1, 3, 1, 0); c_\Pi(u_2^3) = (1, 1, 3, 0, 1); c_\Pi(u_3^3) = (1, 0, 3, 1, 1;);$
$c_\Pi(v_1) = (0, 1, 1, 1, 1); c_\Pi(v_2) = (1, 0, 1, 1, 1); c_\Pi(v_3) = (0, 1, 2, 1, 1).$

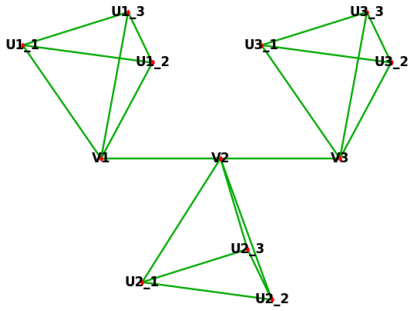

FIGURE 3. A minimum locating coloring of $P_3 \odot C_3$

Case 3 $P_4 \odot C_3$
The partition Π of $V(P_4 \odot C_3)$:
$S_1 = \{u_3^2, v_4\}$
$S_2 = \{u_3^1, u_3^3, v_2\}$
$S_4 = \{u_1^1, u_2^2, u_2^3, u_2^4\}$
$S_5 = \{u_2^1, u_1^2, u_1^3, u_1^4\}$.
Then, the color codes are:
$c_\Pi(u_1^1) = (3, 1, 1, 0, 1); c_\Pi(u_2^1) = (3, 1, 1, 1, 0); c_\Pi(u_3^1) = (3, 1, 0, 1, 1);$
$c_\Pi(u_1^2) = (1, 2, 1, 1, 0); c_\Pi(u_2^2) = (1, 2, 1, 0, 1); c_\Pi(u_3^2) = (0, 2, 1, 1, 1);$
$c_\Pi(u_1^3) = (2, 1, 1, 1, 0); c_\Pi(u_2^3) = (2, 1, 1, 0, 1); c_\Pi(u_3^3) = (2, 1, 0, 1, 1);$
$c_\Pi(u_1^4) = (1, 1, 3, 1, 0); c_\Pi(u_2^4) = (1, 1, 3, 0, 1); c_\Pi(u_3^4) = (1, 0, 3, 1, 1);$
$c_\Pi(v_1) = (2, 0, 1, 1, 1); c_\Pi(v_2) = (1, 1, 0, 1, 1); c_\Pi(v_3) = (1, 0, 1, 1, 1);$
$c_\Pi(v_4) = (0, 1, 2, 1, 1).$

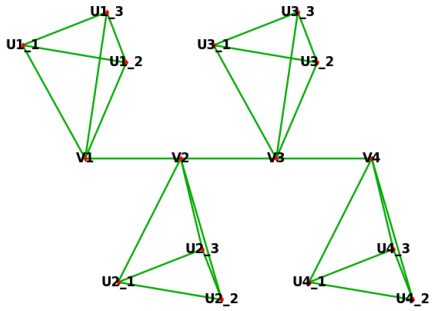

FIGURE 4. A minimum locating coloring of $P_4 \odot C_3$

Case 4 $P_2 \odot C_4$
The partition Π of $V(P_2 \odot C_4)$:

$S_1 = \{u_4^1, u_4^2, v_3\}$
$S_2 = \{u_3^1, u_4^3, v_2\}$
$S_3 = \{u_3^2, u_3^3, v_1\}$
$S_4 = \{u_2^1, u_1^2, u_2^3\}$
$S_5 = \{u_1^1, u_2^2, u_1^3\}$.

Then, the color codes are:
$c_\Pi(u_1^1) = (1,2,1,1,0); c_\Pi(u_2^1) = (2,1,1,0,1); c_\Pi(u_3^1) = (1,0,1,1,2); c_\Pi(u_4^1) = (0,1,1,2,1);$
$c_\Pi(u_1^2) = (1,1,2,0,1); c_\Pi(u_2^2) = (2,1,1,1,0); c_\Pi(u_3^2) = (1,1,0,2,1); c_\Pi(u_4^2) = (0,1,1,1,2);$
$c_\Pi(u_1^3) = (1,1,2,1,0); c_\Pi(u_2^3) = (1,2,1,0,1); c_\Pi(u_3^3) = (1,1,0,1,2); c_\Pi(u_4^3) = (1,0,1,2,1);$
$c_\Pi(v_1) = (1,1,0,1,1); c_\Pi(v_2) = (1,0,1,1,1); c_\Pi(v_3) = (0,1,1,1,1).$

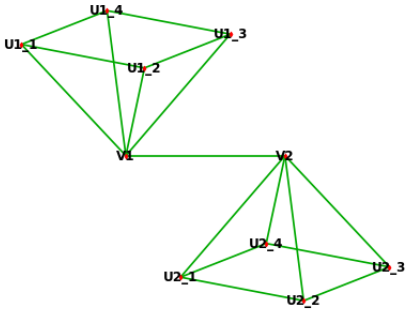

FIGURE 5. A minimum locating coloring of $P_2 \odot C_4$

Case 5 $P_3 \odot C_4$
The partition Π of $V(P_3 \odot C_4)$:

$S_1 = \{u_4^1, u_4^2, v_3\}$
$S_2 = \{u_3^1, u_4^3, v_2\}$
$S_3 = \{u_3^2, u_3^3, v_1\}$
$S_4 = \{u_2^1, u_1^2, u_2^3\}$
$S_5 = \{u_1^1, u_2^2, u_1^3\}$.

Then, the color codes are:
$c_\Pi(u_1^1) = (1,2,1,1,0); c_\Pi(u_2^1) = (2,1,1,0,1); c_\Pi(u_3^1) = (1,0,1,1,2); c_\Pi(u_4^1) = (0,1,1,2,1);$
$c_\Pi(u_1^2) = (1,1,2,0,1); c_\Pi(u_2^2) = (2,1,1,1,0); c_\Pi(u_3^2) = (1,1,0,2,1); c_\Pi(u_4^2) = (0,1,1,1,2);$
$c_\Pi(u_1^3) = (1,1,2,1,0); c_\Pi(u_2^3) = (1,2,1,0,1); c_\Pi(u_3^3) = (1,1,0,1,2); c_\Pi(u_4^3) = (1,0,1,2,1);$
$c_\Pi(v_1) = (1,1,0,1,1); c_\Pi(v_2) = (1,0,1,1,1); c_\Pi(v_3) = (0,1,1,1,1).$

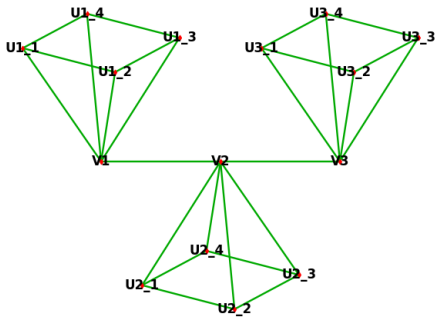

FIGURE 6. A minimum locating coloring of $P_3 \odot C_4$

4. CONCLUSION

Based on the previous discussion, it confirms the effect of graph complexity on the computing process. The more vertices a graph has, the longer it will take to get the *lcn* of the graph, because there are more vertices then there are more combinations of colorings of the graph.

Acknowledgment. We would like to thank the Directorate General of Higher Education, Research and Technology of the Republic of Indonesia for funding this Master's Thesis Grant with contract number 057/E5/PG.02.00.PL/2024.

REFERENCES

[1] G. Chartrand, D. Erwin, M. A. Henning, P. J. Slater, and P. Zhang, "The locating-chromatic number of a graph," *Bull. Inst. Combin. Appl.*, vol. 36, pp. 89–101, 2002.

[2] G. Chartrand, Erwin, M. A. Henning, J. Slater, and P. Zhang, "Graph of order n with locating-chromatic number n-1," *Discrete Math*, vol. 269, pp. 65–79, 2003.

[3] Asmiati, "On the locating-chromatic number of non-homogeneous caterpillars and firecracker graphs," *Far East Journal of Mathematical Sciences*, vol. 100, no. 8, pp. 1305–1316, 2016.

[4] Asmiati, M. Damayanti, and L. Yulianti, "On the locating chromatic number of barbell shadow path graph," *Indonesian Journal of Combinatorics*, vol. 5, no. 2, pp. 82–93, 2021. http://dx.doi.org/10.19184/ijc.2021.5.2.4.

[5] Asmiati, A. Irawan, A. Nuryaman, and K. Muludi, "The locating chromatic number for certain operation of origami graphs," *Mathematics and Statistics*, vol. 11, no. 1, pp. 101–106, 2023. https://doi.org/10.13189/ms.2023.110111.

[6] A. Yulianti, Asmiati, N. Hamzah, and Notiragayu, "The locating chromatic number for amalgamation of some complete graphs," *InPrime: Indonesian Journal of Pure and Applied Mathematics*, vol. 6, no. 1, pp. 76–88, 2024. https://doi.org/10.15408/inprime.v6i1.38711.

[7] A. Irawan, Asmiati, L. Zakaria, and K. Muludi, "The locating-chromatic number of origami graphs," *Algorithms*, vol. 14, no. 6, pp. 1–15, 2021. https://doi.org/10.3390/a14060167.

[8] A. Irawan, Asmiati, B. H. S. Utami, A. Nuryaman, and K. Muludi, "A procedure for determining the locating chromatic number of an origami graphs," *IJCSNS*, vol. 22, no. 9, pp. 31–34, 2022. http://paper.ijcsns.org/07_book/202209/20220905.pdf.

[9] M. Damayanti, Asmiati, Fitriani, M. Ansori, and A. Faradilla, "The locating chromatic number of some modified path with cycle having locating number four," in *Journal of Physics: Conference Series*, vol. 1751, IOP Publishing Ltd, 2021. https://doi.org/10.1088/1742-6596/1751/1/012008.

[10] K. Prawinasti, M. Ansori, Asmiati, Notiragayu, and A. R. G. N. Rofi, "The locating chromatic number for split graph of cycle," in *Journal of Physics: Conference Series*, vol. 1751, IOP Publishing Ltd, 2021. https://doi.org/10.1088/1742-6596/1751/1/012009.

[11] F. Zikra, D. Welyyanti, and L. Yulianti, "The locating-chromatic number of disjoint union of fan graphs," *Jurnal Matematika UNAND*, vol. 11, no. 3, 2022. https://doi.org/10.25077/jmua.11.3.159-170.2022.

[12] B. M. Salindeho, H. Assiyatun, and E. T. Baskoro, "On the locating-chromatic number of subdivisions of friendship graph," *JIMS: Journal of the Indonesian Mathematical Society*, vol. 26, no. 2, pp. 175–184, 2020. https://doi.org/10.22342/jims.26.2.822.175-184.

[13] D. Welyyanti, R. Lestari, and S. R. Putri, "The locating-chromatic number of disconnected graph with path and cycle graph as its components," in *IOP Conference Series*, vol. 1317, IOP Publishing Ltd, 2019. https://doi.org/10.1088/1742-6596/1317/1/012021.

[14] S. Rahmatalia, Asmiati, and Notiragayu, "Bilangan kromatik lokasi graf split lintasan," *Jurnal Matematika Integratif*, vol. 18, no. 1, 2022. https://doi.org/10.24198/jmi.v18.n1.36091.73-80.

[15] R. Frucht and F. Harary, "On the corona of two graphs," *Aequationes mathematicae*, vol. 4, no. 3, pp. 322–325, 1970. https://doi.org/10.1007/BF01844162.

[16] N. Hamzah, Asmiati, and W. D. Amansyah, "Locating chromatic number for corona operation of path and cycle," *Indonesian Journal of Combinatorics*, vol. 8, no. 2, pp. 127–135, 2024. http://dx.doi.org/10.19184/ijc.2024.8.2.6.