Implementation of the greedy algorithm on graph coloring

T N Sipayung\textsuperscript{1}, S Suwilo\textsuperscript{2}, P Gultom\textsuperscript{2} and Mardiningsih\textsuperscript{2}

\textsuperscript{1}Graduate School of Mathematics, Universitas Sumatera Utara, Medan, Indonesia
\textsuperscript{2}Department of Mathematics, Universitas Sumatera Utara, Medan, Indonesia

*Corresponding author
E-mail: tetty_sipayung@ust.ac.id

Abstract. Graph theory is part of the field of mathematics that can be applied in various other fields of science to solve problems. One of them is the problem of determining the color on the map. The map that will be colored here is a map of the Deli Serdang regency which consists of 22 sub-districts. The coloring of the map is done by first modeling it in the form of a graph. One way to determine the minimum color of a graph is to use a greedy algorithm. From the map, we get a dual graph with 22 vertices and 41 edges. Based on the greedy algorithm that has been applied, the minimum number of colors is obtained as many as four colors, namely blue, green, red, and yellow with each district directly bordering having a different color. The results of map coloring by applying the greedy algorithm are also obtained with the help of the python 3.7 programming language.

1. Introduction

In modeling the problem, it can be done through graphs and its implementation in graph coloring. The graph coloring issue is one of the most interesting and challenging algorithms for solving in computer science, mathematics, and operations research [9]. The problem of graph coloring can of course be connected with the field of cartography. Cartography is the art of making maps.

Topographic maps make a landscape a significant visual impression and illusion [4]. Maps and geographic information (GI) require significant skills because they can connect and combine sets of data based on their geographic location, and present information in a visually appealing and factual way [13]. Several previous studies related to maps include coloring the map of a city by applying the four-color theorem [12], the ant algorithm [5] and depth first search algorithm [7]. But in this paper, we explain the coloring of the graph using a greedy algorithm assisted by the python programming language. Before discussing the application of the greedy algorithm, several basic definitions related to graphs and graph coloring are presented.

An ordered triple $G = (V(G), E(G))$, $I_G$ is a graph, where $V(G)$ is a nonempty set. $V(G)$ and $I_G$ are disjoints, while $E(G)$ is a disjoint from $V(G)$. $I_G$ is an incidence relation that relates an unordered pair of items (same or distinct) of $V(G)$ with every element of $E(G)$. The vertices of $G$ are elements of $V(G)$, whereas the edges of $G$ are elements of $E(G)$. The vertex and edge sets of $G$ are $V(G)$ and $E(G)$, respectively. If, we put $I_G(v) = uv$ in the edge $e$ of $G$ [3]. Every graph has an adjacency matrix, which is a binary $n \times n$ matrix $A$ in which $a_{ij} = 1$ and $a_{ji} = 1$ if vertex $v_i$ is next to vertex $v_j$, and otherwise $a_{ij} = 0$ and $a_{ji} = 0$ [11]. The number of linkages
occuring to the vertex determines the degree of a node in a graph. \( \text{Deg}(u) \) describes the degree of a vertex \( u \) \[12\]. A map \( C : V \rightarrow \{ C_1, C_2, \ldots, C_n \} \) is a coloring of \( G = (V,E) \). \( C(v_i) \neq C(v_j) \forall v_i,v_j \in E \). Alternatively, the coloring of a graph \( G = (V,E) \) is a mapping \( c : v \rightarrow s \), where \( s \) is a finite set a color, such that if \( vw \in E \), \( c(v) \neq c(w) \) \[1\]. The chromatic number of \( G \), written \( \chi(G) \)[10] is the lowest feasible number of colors used to solve the problem. Some of these basic definitions are needed in understanding the application of the greedy algorithm.

A greedy approach can be used to color the graph \( G \) with less colors by starting with the enumeration of fixed vertices \( v_1,\ldots,v_n \) of \( G \). Consider each vertex individually then color each \( v_i \) with the first available color, such as the smallest positive integer that hasn’t been used to color \( v_i \)’s neighbors among \( v_1,\ldots,v_{i-1} \) \[6\]. Greedy techniques rarely produce optimum answers in many situations, but greedy heuristics can produce local optimal solutions that are near to global optimal solutions in an acceptable amount of time \[8\].

Graph coloring problems are widely used to solve computer-based applications and problems \[2\]. Therefore, to find out the results of the implementation of the greedy algorithm, it is assisted by using the python programming language.

2. Methodology
This research method is a case study using a sub-district map in Deli Serdang Regency which consists of 22 sub-districts.

How the greedy algorithm works in the case of map coloring will be described as follow:

1. Initialize the solution set with empty.
2. Selection of the node to be filled with color with the node selection function.
3. Select a color candidate using the color selection function. Subtract the color in the candidate set \( C \), if the color is taken from the candidate set \( C \).
4. Sort the vertices from the one with the highest number of edges to the one with the fewest.
5. Check the suitability of the selected color using the eligibility function. The appropriate color used for the selected node will be included in the solution set, if it is not feasible then the process returns to step 2.
6. Check whether the solution includes coloring all vertices with an optimal solution with an objective function. The coloring process will stop if the coloring has obtained an optimal solution, if it is not optimal, return to step 2.

Before the greedy algorithm is applied, first every sub-district in Deli Serdang regency is labeled with a vertex. This can be seen in the table bellow:

| District Name | Vertex | District Name | Vertex |
|---------------|--------|---------------|--------|
| Gunung Meriah | \( v_1 \) | STM Hilir   | \( v_{12} \) |
| STM Hulu      | \( v_2 \) | Patumbak    | \( v_{13} \) |
| Bangun Purba  | \( v_3 \) | Biru-biru   | \( v_{14} \) |
| Galang        | \( v_4 \) | Deli Tua    | \( v_{15} \) |
| Pagar Merbau  | \( v_5 \) | Namorambe   | \( v_{16} \) |
| Lubuk Pakam   | \( v_6 \) | Sibolangit  | \( v_{17} \) |
| Beringin      | \( v_7 \) | Katalim Baru| \( v_{18} \) |
| Pantai Labu   | \( v_8 \) | Pancur Batu| \( v_{19} \) |
| Percut Sei Tuan | \( v_9 \) | Sunggal    | \( v_{20} \) |
| Batang Kuis   | \( v_{10} \) | Hamparan Perak | \( v_{21} \) |
| Tanjung Morawa | \( v_{11} \) | Labuhan Deli | \( v_{22} \) |
The map of the sub-districts in Deli Serdang district can be seen in figure 1. Each point represents a sub-district. The points are connected by a line but specifically the area that is next to each other so that the graph model can be seen in figure 2.

![Figure 1. Map of districts in Deli Serdang regency.](image1)

![Figure 2. Graph model of district map in Deli Serdang regency.](image2)

Based on the graph model in figure 2, the degree of each point can be determined. The degrees of each point are arranged from the highest to the lowest level as shown in the following table:

| Vertex | Degree | Vertex | Degree |
|--------|--------|--------|--------|
| $v_{11}$ | 8      | $v_3$   | 3      |
| $v_{12}$ | 7      | $v_5$   | 3      |
| $v_{14}$ | 5      | $v_6$   | 3      |
| $v_{17}$ | 5      | $v_8$   | 3      |
| $v_7$   | 5      | $v_9$   | 3      |
| $v_{10}$ | 4      | $v_{15}$ | 3     |
| $v_{13}$ | 4      | $v_{18}$ | 3     |
| $v_{16}$ | 4      | $v_{20}$ | 3     |
| $v_{19}$ | 4      | $v_{21}$ | 2     |
| $v_4$   | 4      | $v_1$   | 1      |
| $v_2$   | 3      | $v_{22}$ | 1     |
3. Results
Through several things that have been done, the greedy algorithm is applied. The following are
the stages of map coloring using the greedy algorithm by python:

**Input**: vertex data of a graph which is arranged from node of highest degree to lowest.

**Output**: colored map.

The application of the greedy algorithm for coloring the sub-district map in Deli Serdang
district with python language requires four iterations, namely:

**iteration 1**: install necessary packages.

**iteration 2**: initialize the graph and greedy algorithm for coloring function.

**iteration 3**: read map file and check the sample records.

**iteration 4**: visualize the coloring results using map data and the plotlib library.

The output of the implementation of the greedy algorithm with the python programming
language is shown in figure 3. While the map of the districts in Deli Serdang Regency which is
colored in four colors along with a description of each district is shown in figure 4 below.

![Figure 3. Map coloring for Deli Serdang regency.](image)

![Figure 4. Colored map of the district in Deli Serdang.](image)

4. Conclusion
Map coloring carried out in sub-districts in Deli Serdang regency using the greedy algorithm,
obtained several conclusions, namely: there are 22 vertex representing sub-districts in Deli
Serdang regency. By interpreting the map into a graph model, one of the alternatives generated
is that there are 41 edges formed. There is a set of $C$ which consists of eight colors, namely
blue, green, red, yellow, purple, pink, orange, and chocolate. Tanjung Morawa sub-district has the highest number of point degrees so that area is first colored in blue. Based on the greedy algorithm with the help of the python 3.7 programming language, four colors were obtained in coloring the sub-district map in Deli Serdang Regency, namely blue, green, red, and yellow.

Acknowledgment
The author would like to thank the Universitas Katolik Santo Thomas based on the letter of agreement for further study permits number 0476/YST/G.35/04’21 for the support provided both materially and spiritually so that this paper can be published through this scientific article.

References
[1] Al-Omari H and Khair E S 2006 New graph coloring algorithms American Journal of Mathematics and Statistics 2(4) pp 739-741.
[2] Aslan M and Nurdan A B 2016 A performance comparison of graph coloring algorithms International Journal of Intelligent Systems and Applications in Engineering 4(Special Issue) pp 1-7.
[3] Balakrishnan R and Ranganathan K 2012 A textbook of graph theory Second Edition Springer: New York.
[4] Carbonell C and Carrera 2018 2D cartography training: has the time come for a paradigm shift? International Journal of Geo-Information 7(5) pp. 1-13 doi:10.3390/ijgi7050197.
[5] Chen H and Zhou P 2013 An ant algorithm for solving the four-coloring map problem Ninth International Conference on Natural Computation (ICNC) IEEE pp 491-495 doi:10.1119/ICNC.2013.6818026.
[6] Diestel R 2000 Graph theory Electronic Edition 2000 Springer-Verlag: New York.
[7] Gao F et al 2012 An algorithm for map coloring problem based on depth first search Advanced Materials Research 424-425 pp 480-483 doi: 10.4028/www.scientific.net/AMR.424-425.480.
[8] Jain A M S and Manohar K 2015 Greedy algorithm Journal of Advance Research in Computer Science and Engineering 2(4) pp 11-15.
[9] Meraihi Y 2020 A Modified binary crow search algorithm for solving the graph coloring problem International Journal of Applied Evolutionary Computation 11(2) pp 28-46 doi: 10.4018/IJAECS.202004103.
[10] Ouerfelli L and Bouziri H 2011 Greedy algorithms for dynamic graph coloring International Conference on Communications, Computing and Control Applications (CCA) doi: 10.1109/cca.2011.6031437.
[11] Sarma S V M 2012 Application of graph theory in human life International Journal of Computer Application 2(2) pp 21-30.
[12] Stojkovikj N et al 2020 Application of the four-color theorem for coloring a city map Balkan Journal of Applied Mathematics and Informatics (BJAMI) 3(1) pp 25-35.
[13] Virrantaus K et al 2009 ICA research agenda on cartography and GI science The Cartographic Journal 46(2) pp 63-75 doi:10.1179/000870409X459824.