InPrime: Indonesian Journal of Pure and Applied Mathematics,
Vol. 4, No. 1 (2022), pp. 24 – 32, doi: 10.15408/inprime.v4i1.24327
p-ISSN 2686-5335, e-ISSN: 2716-2478

Application of Genetic Algorithm on Inclusive Labeling of a Graph

Kiswara Agung Santoso*, Bagus Arief Setiawan and Kusbudiono
Department of Mathematics, Jember University, Jl. Kalimantan 37, Jember, Indonesia
Email: kiswaras@gmail.com

Abstract

As science developed, heuristic methods began to be used in graph coloring. Heuristic methods that have been used for graph coloring include Genetic Algorithm, Tabu Search, and Ant Colony Algorithm. A Genetic Algorithm is a method for solving optimization problems. In this study, the Genetic Algorithm will be used for the issue of labeling irregular vertices of inclusive distances to label any graph inclusively. We restrict an inclusive 1-distance to a simple graph using one-point crossover and mutation. The steps are a generation of random chromosomes, evaluating chromosome fitness values with tournament selection, conducting an evolutionary process consisting of one-point crossover and mutation, repeating the process until the termination criteria are met. The results of implementing the genetic algorithm on inclusive labeling can be determined by the chromatic number based on the adjacency matrix. The results of this labeling can be used as an alternative solution to the problem of inclusive labeling.

Keywords: Genetic Algorithm; graph labeling; inclusive labeling.

Abstrak

Seiring berkembangnya ilmu pengetahuan metode heuristic *mulai digunakan dalam pewarnaan graf. Metode heuristic yang telah digunakan untuk pewarnaan graf antara lain Algoritma Genetika,* Tabu Search*, dan Algoritma Semut* (Ant Colony). *Algoritma Genetika merupakan metode untuk menyelesaikan masalah optimasi. Pada penelitian ini, Algoritma Genetika digunakan untuk masalah pelabelan titik tak-teratur jarak inclusive agar dapat melabeli sebarang graf secara inclusive. Kami membatasi lingkup penelitian dengan menerapkan jarak* inclusive *1 pada graf sederhana, menggunakan crossover satu titik dan mutasi. Metode yang digunakan dalam penelitian ini adalah studi literatur dengan mengkaji penggunaan Algoritma Genetika pada pelabelan titik tak-teratur jarak inclusive suatu graf. Langkah-langkah yang dilakukan adalah: pembangkitan kromosom secara acak, evaluasi nilai fitness kromosom dengan* tournament selection*, melakukan proses evolusi yang terdiri dari* crossover *satu titik dan mutasi, perulangan proses sampai kriteria pemerhentian terpenuhi. Hasil implementasi algoritma genetika pada pelabelan inclusive adalah dapat mengetahui bilangan kromatik berdasarkan matriks* adjacency. *Hasil pelabelan ini dapat dijadikan sebagai salah satu alternatif penyelesaian masalah pelabelan inklusif.*

Kata Kunci : *Algoritma Genetika*; *pelabelan graf*; *pelabelan inklusif.*

1. INTRODUCTION

Labeling is one part of Graph Theory. Graph labeling was first introduced by Sadlàčk [1], then Stewart [2], Kotzig and Rosa [3]. Graph labeling is a function that maps elements of a graph to a specific set of integers. Graph labeling is grouped into vertex labeling, edge labeling, and total labeling (vertex and edge) based on the elements. At present, graph labeling has been widely developed along with the development of science. Slamin [4] introduced a new type of irregular labeling, called on-distance vertex irregular labeling. The labeling evaluation is based on the surrounding elements adjacent to the reference vertex (the vertex to be labeled). Almost the same concept was introduced by [5], namely on inclusive distance vertex irregular labeling. The difference in the labeling concept is

*) Corresponding author

Submitted January 20th, 2022, Revised March 23rd, 2022, Accepted for publication March 26th, 2022.

based on how to calculate the weight. I inclusive distance vertex irregular labeling is a vertex labeling with the weight of the point $v \in V(G)$ is the sum of the labels of the vertex v and all the labels of the vertex that are 1-distance from the vertex v.

As science developed, heuristic methods began to be used in graph coloring. The heuristic method is a solution using the concept of an interactive approach to producing a solution that is close to optimal [6]. Heuristic methods that have been used for graph coloring include Genetic Algorithm, Tabu Search, and Ant Colony Algorithm.

A Genetic Algorithm is a dynamic random searching algorithm used to solve optimization problems. John Holland invented it in the 1960s. The Genetic Algorithm is based on the biological methods to produce the solutions (next population) [7]. Modeling is carried out in the form of an array to apply the Genetic Algorithm to graph problems. We encode the graph into an array based on the adjacency matrix. Genetic Algorithms are stochastic search methods that are applied widely in engineering, medicine, and machine learning problems [7]. In previous research, Genetic Algorithms have been commonly used in graph theory. Nirmala and Ramprasad [8] applied Genetic Algorithms to Graph Theory. Hindi and Yampolskiy [9] applied Genetic Algorithms to Graph Coloring Problems. Barod, Hawanna, and Jagtap [10] applied Genetic Algorithms and Memetic Algorithm to Graph Coloring. Assi, Halawi, and Haraty [11] say that the Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem. Anggraini, Rosyida and Asih [12] Graph Coloring Problem Solution with Genetics Algorithm. Based on this, we use a Genetic Algorithm for the problem of an inclusive distance vertex irregular labeling so that it can label any graph inclusively. We restrict our research by applying an inclusive 1-distance to a simple graph, using one-point crossover and mutation.

2. METHODS

The method used in this research is a literature study by examining the use of Genetic Algorithms in inclusive distances vertex irregular labeling of a graph [13]. After reviewing the application of the Genetic Algorithm for graph labeling, the implementation of the Genetic Algorithm in graph labeling will be used as a problem that will be studied more deeply. The steps taken in solving the graph labeling problem using the Genetic Algorithm are: generation of random chromosomes assuming the graph has become an adjacency matrix, finding the chromosome fitness value with tournament selection, carrying out an evolutionary process consisting of one-point crossover and mutation, looping until termination criteria are met. The final result is a graph labeling with chromatic numbers and no neighboring vertices with the same weight.

The following method is to create a program with MATLAB. At the end of our research, we present the results and conclusions to analyze the solution to our problems. Analysis of the results carried out with the criteria is considered good if the results obtained are close to or equal to the actual results.

3. RESULTS AND DISCUSSIONS

This section will discuss the implementation of genetic algorithms on several graphs based on previous research. The following will present the results of the performance of the genetic algorithm. Solving the inclusive vertex labeling problem using a genetic algorithm is done by modeling the graph into the form of an adjacency matrix. The process starts with chromosome construction, evaluation of fitness values, selection, crossover, and mutation using a matrix basis. Implementing the genetic algorithm is a repetition of the entire process above. So it is necessary to explain the process.

Table 1. Results of Genetic Algorithm Implementation [5].

Previous Research		**Genetic Algorithm**
Graph Type	$\widehat{dis}$	
S_1	∞	-
S_2	2	1 1 2
S_4	4	2 1 2 4 3
S_7	7	3 2 4 3 5 1 7 6
S_{11}	11	1 11 1 3 9 2 4 10 8 7 6 5
P_2	∞	-
P_3	2	1 1 2
P_5	3	3 2 1 1 1
P_8	3	1 1 1 2 2 3 3 3
P_{12}	5	1 1 4 5 5 5 2 4 3 1 1 2
C_3	∞	-
C_4	4	3 4 2 1
C_6	3	1 1 1 2 3 3
C_9	4	1 2 2 3 4 3 4 1 1
C_{13}	5	1 1 1 3 3 2 5 5 5 4 4 3 2
$B_{r3,5}$	5	4 4 1 2 3 1 4 5
$B_{r3,6}$	6	6 3 1 1 6 4 2 3 5
$B_{r3,7}$	7	6 7 1 2 1 6 4 3 5 7
$B_{2,4}$	4	1 1 3 2 4 3 3 3 3 4 2
$B_{2,5}$	5	1 1 3 2 4 2 5 5 5 4 2 5 1
$B_{2,6}$	6	6 6 4 1 2 3 5 6 4 1 5 3 2 1 4
$F_{3,3}$	4	2 3 4 4 3 1 4 4 1 1 2 4
$F_{4,3}$	5	5 1 5 1 5 1 3 4 5 3 5 4 5 5 5 4
$F_{5,3}$	6	1 2 1 6 2 2 6 6 4 3 2 6 6 5 6 5 3 2 6 6
$C_{6,2}$	7	5 7 1 7 3 6 1 4 2 7 1 7 3 6 1 6 1 5
$C_{7,2}$	8	7 7 6 1 2 1 6 4 6 2 3 5 7 4 1 8 7 8 3 6 3
$C_{8,2}$	9	6 9 1 8 5 1 6 5 8 4 1 8 8 2 9 2 1 2 9 7 9 3 3 5

3.1 Modeling a Graph into an Adjacency Matrix

The input used in this research is the adjacency matrix of a graph which is defined as follows:

$$a_{ij} = \begin{cases} 1, & \text{if there is an edge } \{u, v\} \\ 0, & \text{if there is no edge} \end{cases}$$

Example:

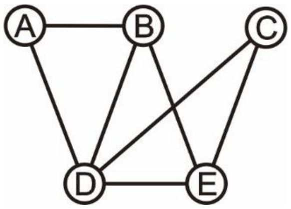

Figure 1. Graph $\boldsymbol{G}$.

The following is the adjacency matrix of graph G (M) from Figure 1:

$$M = \begin{bmatrix} & A & B & C & D & E \\ A & 0 & 1 & 0 & 1 & 0 \\ B & 1 & 0 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 1 & 1 \\ D & 1 & 1 & 1 & 0 & 1 \\ E & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The adjacency matrix will then be modified to calculate the weight of inclusive labeling. The following will present the results of the modification of the adjacency matrix:

$$M = \begin{bmatrix} & A & B & C & D & E \\ A & 1 & 1 & 0 & 1 & 0 \\ B & 1 & 1 & 0 & 1 & 1 \\ C & 0 & 0 & 1 & 1 & 1 \\ D & 1 & 1 & 1 & 1 & 1 \\ E & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

3.2 Chromosome Construction Based on Adjacency Matrix and Evaluation of Fitness Value

The first process is defining the number of chromosomes that have been determined at the beginning of the program, i.e., four chromosomes. These chromosomes will later represent the labels of the vertices on the graph.

Chromosome 1 :	2	2	1	2	2
Chromosome 2 :	1	1	2	2	2
Chromosome 3 :	1	2	2	1	2
Chromosome 4 :	1	1	2	2	2

Each chromosome will be multiplied by the adjacency matrix to get the inclusive labeling weight value. The following is an illustration of the calculation of the weight value, which will later determine the fitness value:

$$\begin{array}{|c|c|c|c|c|}\hline 2 & 2 & 1 & 2 & 2 \\ \hline \end{array} \cdot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{array}{|c|c|c|c|c|}\hline 6 & 8 & 5 & 9 & 7 \\ \hline \end{array}$$

The product of the adjacency matrix of graph G with one of the chromosomes is obtained by the weight value $6 \quad 8 \quad 5 \quad 9 \quad 7$. Fitness values are defined based on two things, i.e., the label that appears and the similarity of the number of weights. The following is the calculation of the fitness value from the label results above:

$$\frac{1}{f(v) + N * p(v)} = \frac{1}{2 + 5 \times 0} = \frac{1}{2} = 0.5.$$

with the largest label $f(v)$, which is 2, several vertex N, which is 5, and labeling error / the same number of weights $p(v)$. Because no vertex has the same weight, $p(v)$ is 0. If there is a label error with the same value vertex weights, the labeling is inappropriate. Based on this, the resulting fitness value will be smaller when weights are similar. The smaller the fitness value, the smaller the probability of selecting the chromosome. This is because the best chromosome will be selected in the selection process.

Chromosome Fitness Value 1: 0.50,
Chromosome Fitness Value 2: 0.14,
Chromosome Fitness Value 3: 0.50,
Chromosome Fitness Value 4: 0.14.

3.3 Evolution Process

Each fitness value will be selected so that the best chromosome is obtained. The selection process is carried out using the tournament selection method. This selection will choose and compare the fitness value of each chromosome so that the two most significant values are obtained in each generation. These two chromosomes will be used in the evolutionary process.

Chromosome Fitness Value 1: 0.50,
Chromosome Fitness Value 3: 0.50.

The evolution process in the Genetic Algorithm consists of crossover and mutation. Crossover is done to get a new chromosome. The new chromosomes are obtained by crossing parts of each chromosome that have gone through the previous selection process. The crossover process is carried out at a predetermined point without a random process. This research's selection of crossover points is based on positive integers from the mean chromosome value. The following will present the crossover process with two predetermined chromosomes.

Example:

Chromosome 1 :	2	2	1	2	2
				↕	
Chromosome 2 :	1	2	2	1	2

The first step is to determine the crossover point. As in the example of the chromosome consisting of 5 genes, a crossover point is obtained 3. The selection of the crossover point does not use a random process because the random process allows the chromosomes got to be more comprehensive. This can result in a longer computation time required. Based on this, the chromosome will be divided into two parts at a predetermined crossover point. Furthermore, each part of the chromosome will be crossed with another chromosome. The results of this crossover process are presented below.

Chromosome 3 :	2	2	1	1	2
Chromosome 4 :	1	2	2	2	2

Unlike the crossover process, mutations are carried out in each gene based on a predetermined mutation rate. Mutations are carried out provided that if the generated random value is less than equal to the mutation rate, the gene will be mutated.
Example:

2	2	1	1	2

For example, generate any random number for each gene in sequence. When the first gene's random value is x and is more than equal to the mutation rate, then the value of the first gene will not be mutated. This also applies if the random value in the fourth gene y is less than equal to the mutation rate, then the value of the fourth gene will be mutated so that it is obtained as follows.

2	2	1	2	2

The entire process above will continue to be carried out until it meets the termination criteria. This research uses the termination criteria: the number of generations and convergent solutions. Figure 2 shows that the labeling of graph G has converged at a value of 0.5. Therefore, the process will be stopped. The following presents the results of the simulation of the implementation of the genetic algorithm.

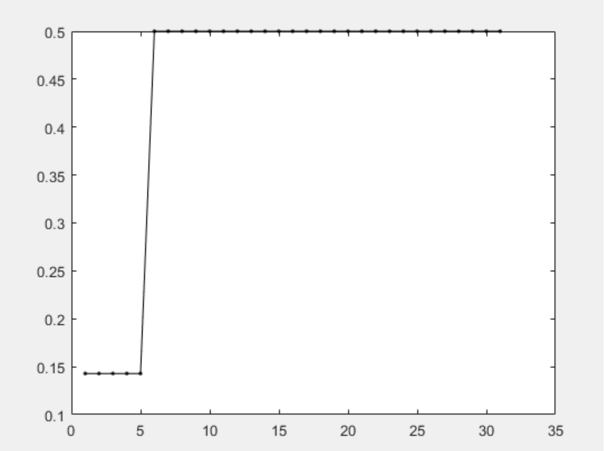

Figure 2. *G* Fitness Score Chart.

3.4 Simulation

1. Generating Chromosomes
   Chromosomes will be generated randomly from numbers 1 to 2 based on the lower bound value. Chromosomes have generated a total of 4 based on the provisions of the researcher. Table 2 presents the initial generation of chromosomes.

Table 2. Initial Chromosomes.

i	Chromosomes (v)				
1	2	1	2	2	2
2	1	1	1	2	1
3	1	1	1	2	2
4	1	1	2	2	2

2. Evaluation of Fitness Value
Chromosomes will be evaluated for many labels and weighing errors based on the chromosome product with the adjacency matrix on the graph. Table 3 presents the evaluation process.

Table 3. Initial Chromosomal Evaluation.

i	Chromosomes (v)					Labels f(v)	Weighting Error p(v)	Fitness Value
1	2	1	2	2	2	2	1	0.14
2	1	1	1	2	1	2	2	0.08
3	1	1	1	2	2	2	1	0.14
4	1	1	2	2	2	2	1	0.14

3. Selection
The selection process is carried out by selecting each generation's most significant fitness value. Selection results will be presented in Table 4.

Table 4. Chromosomal Selection.

Selection Result	Chromosomes					Fitness Value
1	2	1	2	2	2	0.14
3	1	1	1	2	2	0.14

4. Crossover
The crossover process is carried out with a one-point crossover. The crossover point is selected based on the gene mean by rounding off. Table 5 presents the crossover process.

Table 5. Crossover Process

Parent Chromosomes					Crossover Result				
2	1	2	2	2	2	1	2	2	2
1	1	1	2	2	1	1	1	2	2

5. Mutation
The mutation process is carried out to change a gene with a random number between 1 to 2 based on the lower bound. Mutations are carried out with a predetermined mutation rate. Table 6 presents the mutation process.

Table 6. Mutation Process.

Parent Chromosomes					Mutation Result				
2	1	2	2	2	2	1	2	2	2
1	1	1	2	2	1	1	2	2	2

6. Evaluation of Termination Criteria

The new chromosomes resulting from the evolutionary process will be recombined with the parent chromosomes. After forming a new population, we evaluate whether the evolutionary process can continue or stop. The results of chromosome regeneration are presented in Table 7.

Because the minimum labeling with zero weighting error has not been found, the evolution process will continue to be repeated from selection, crossover, and mutation. The evolution process stops at the 30th generation, with the fitness value obtained has converged as much as 25. The final result of the evolution process is four chromosomes with a fitness value of 0.50. The following are the evolutionary process results, which will be presented in Table 8.

Table 7. Evaluation of Early Chromosomal Regeneration.

i	Chromosomes (v)					Labels f(v)	Weighting Error p(v)	Fitness Value
1	1	1	1	2	2	2	1	0.14
2	1	1	2	2	2	2	1	0.14
3	2	1	2	2	2	2	1	0.14
4	1	1	2	2	2	2	1	0.14

Table 8. The 30th Generation Chromosome Results.

i	Chromosomes (v)					Labels f(v)	Weighting Error p(v)	Fitness Value
1	1	2	2	2	2	2	0	0.50
2	1	2	2	2	2	2	0	0.50
3	1	2	2	2	2	2	0	0.50
4	1	2	2	2	2	2	0	0.50

Based on the results above, four labels are considered feasible/entire, with all the labels having the same genes. Implementing the genetic algorithm on graph G shows that the largest k value/label is two so that $\widehat{dis}(G) = 2$. If we review based on previous research, we get $\widehat{dis}(G) = 2$. This indicates that the evolution results provide an appropriate labeling value because they get the same $\widehat{dis}(G)$ value.

4. CONCLUSIONS

The results of the implementation of the genetic algorithm on inclusive labeling can be seen in the chromatic number, which follows previous research. The results of this labeling are based on the adjacency matrix, and there is no equal weight on the labels. The results of this implementation can be used as an alternative solution to the inclusive labeling problem.

We suggest applying genetic algorithms to labeling other graphs with any n vertex or complex graphs for further research. Further research can use different algorithms such as tabu search, ant colony, or others for inclusive labeling.

REFERENCES

[1] J. Sedlacek, "Problem 27," in *Theory of Graphs and it's Applications (Smolenice, 1963)*, Publ. House Czechoslovak Acad. Sci., Prague, 1964, pp. 163-164.

[2] B. M. Stewart, "Magic graphs," *Can. J. Math,* vol. 18, pp. 1031-1059, 1966.

[3] A. Kotzig and A. Rosa, "Magic Valuations of Finite Graph," *Canad. Math. Bull.,* vol. 13, pp. 451-461, 1970.

[4] Slamin, "On Distance Irregular Labelling of Graphs," *Far East Journal of Mathematical Sciences (FJMS),* vol. 102, no. 5, pp. 919-932, 2017.

[5] M. Bača, A. S. Feňovčíková, Slamin and K. A. Sugeng, "On Inclusive Distance Vertex Irregular Labelings," *Electronic Journal of Graph Theory and Applications,* vol. 6, no. 1, pp. 61-83, 2018.

[6] B. Şahin and A. Şahin, "On Domination Polynomials of Caterpillar Graphs," *Turk. J. Math. Comput. Sci.,* vol. 9, p. 34–38, 2018.

[7] H. Farooq, M. N. Zakaria, M. F. Hassan, and S. Sulaiman, "An Interactive Visualization of Genetic Algorithm on 2-D Graph," *International Journal of Software Science and Computational Intelligence,* vol. 4, no. 1, pp. 34-54, 2020, DOI: 10.4018/jssci.2012010102.

[8] G. Nirmala and D. Ramprasad, "Application of Genetic Algorithm in Graph Theory," *International Journal of Scientific and Research Publications,* vol. 2, no. 7, pp. 1-3, 2012.

[9] M. M. Hindi and R. V. Yampolskiy, "Genetic Algorithm Applied to the Graph Coloring Problem," in *Proceedings of the Twenty-third Midwest Artificial Intelligence and Cognitive Science Conference, April 21 – 22,* pp. 60-66, University of Cincinnati, Ohio, 2012.

[10] P. Barod, V. Hawanna and V. Jagtap, "Genetic Algorithm and Memetic Algorithm on Graph Colouring," *Scientific Journal of Impact Factor (SJIF),* vol. 1, pp. 269-276, 2014.

[11] M. Assi, B. Halawi and R. A. Haraty, "Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem," *Procedia Computer Science,* vol. 126, p. 899–906, 2018.

[12] L. A. Anggraini, I. Rosyida and T. S. N. Asih, "Graph Coloring Problem Solution with Genetics Algorithm," *UNNES Journal of Mathematics,* vol. 8, no. 1, pp. 30-39, 2019.

[13] A. R. Savitri, I. Halikin and K. Wijaya, "On Inclusive 1-Distance Vertex Irregularity Strength of Firecracker, Broom, dan Banana Tree," in *Proceeding of International Conference on Mathematics and Islam (ICMIs),* pp. 228-232, Mataram, 2018.