International Journal of Civil Engineering and Technology (IJCIET)
Volume 9, Issue 9, September 2018, pp. 1755–1765, Article ID: IJCIET_09_09_170
Available online at http://www.iaeme.com/ijciet/issues.asp?JType=IJCIET&VType=9&IType=9
ISSN Print: 0976-6308 and ISSN Online: 0976-6316

A REVIEW OF PRIM AND GENETIC ALGORITHMS IN FINDING AND DETERMINING ROUTES ON CONNECTED WEIGHTED GRAPHS

Mhd. Furqan
Postgraduate School of Computer Science, Universitas Sumatera Utara, Medan, Indonesia
Department of Computer Science, Universitas Islam Negeri Sumatera Utara, Medan

Herman Mawengkang
Department of Mathematics, Universitas Sumatera Utara, Medan, Indonesia

Opim Salim Sitompul
Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia

A.P.U. Siahaan, Muhammad Donni Lesmana Siahaan
Faculty of Science and Technology, Universitas Pembangunan Panca Budi, Medan, Indonesia

Wanayumini
Postgraduate School of Computer Science, Universitas Sumatera Utara, Medan, Indonesia

Nazaruddin Nasution
Department of Physics, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia

Sriani
Department of Computer Science, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia

ABSTRACT

Searching for the shortest route when traveling is something that needs to be done in addition to finding the destination city. The reason for finding the shortest route is to summarize the trip and save travel costs. Another issue in making a useful trip is city tracing done by a salesperson, where a salesperson must visit several places to distribute goods, so he will only stop at that place once until the last place to be reached is reached until returning to his place of origin. The Prim algorithm is used to find the minimum generator tree from a weighted connected graph by taking the side/segment that has the smallest weight of the graph, where the line segment is side by side with the stretched tree that has been created, and that does not form a cycle.

Genetic algorithms are also a method used to solve an optimal solution search problem in an optimization problem. This algorithm follows the genetic process of biological organisms based on Charles Darwin's theory of evolution. The implementation of genetic algorithms can be tested in finding the optimal route solution on a graph. Based on the results of genetic testing algorithms have different results on each test. It is because genetic algorithms use random numbers to generate probabilities for each generation. The results of the research describe Prim's algorithm better than Genetics because it has consistent results. It is not seen in the Genetic algorithm that expects opportunities from random numbers to appear. However, Genetic algorithms are excellent algorithms when applied to extensive cases because this algorithm will determine the optimal route of all possible routes.

Key words: Prim, Genetic Algorithm, Minimum Spanning Tree, Artificial Intelligent

Cite this Article: Mhd. Furqan, Herman Mawengkang, Opim Salim Sitompul, A.P.U. Siahaan, Muhammad Donni Lesmana Siahaan, Wanayumini, Nazaruddin Nasution, Sriani, A Review of Prim and Genetic Algorithms in Finding and Determining Routes on Connected Weighted Graphs. *International Journal of Civil Engineering and Technology*, 9(9), 2018, pp. 1755-1765.
http://www.iaeme.com/IJCIET/issues.asp?JType=IJCIET&VType=9&IType=9

1. INTRODUCTION

Every communication equipment requires electrical energy to operate smoothly [1]–[5] Speed and The faster the data transmission process, the smoother the data exchange. Speed requires an excellent algorithm to work optimally [6]–[8]. Spanning trees are the total weight of all edges in the tree, and it is not uncommon to span more than one tree. The term Minimum spanning tree refers to a spanning tree that has the lowest complexity among all spanning trees. As with the spanning tree in general, the minimum stretch tree can also amount to more than one. The Minimum Spanning Tree is applied directly in network design and is also used to formulate solutions to traveling salesmen problem [9][10].

Traveling Salesman Problem (TSP) is illustrated as a salesman trip to determine the path taken from a city to pass through all cities and return to the original city, provided that each city can only be passed in one trip. Determining the trajectory of this case is one of the problems of the graph, how to form a circuit to pass through all vertices with the total weight of the minimum circuit-forming side. The weight on the side that connects a pair of vertices represents time, cost, and distance. The method that can be used to solve TSP is the Prims method and Genetic algorithm [11]. If G is a weighted graph, then Minimum Spanning Tree of G is Spanning Tree with the smallest amount of weight [12]. The Prims and Genetics algorithms can be performed to get the Minimum Spanning Tree route. Both of these algorithms both seek the shortest distance from the entire route. It is done by preventing graphs from forming cycles. The difference is, Prims Algorithm ranks its weight from large to small and executes the most significant weight first [13] while Genetic algorithm processes opportunities by using random numbers that appear during generation processes [14]. TSP and MST are expected to be completed by both Prim and Genetic algorithms.

2. THEORIES

2.1. Minimum Spanning Tree

The minimum cost of a stretched tree is a stretched tree, only the weight or length associated with the sides of the tree is the minimum. Examples of cases where we have a group of

islands, and we want each island to be connected to a bridge as a link so that we are possible for someone to travel from one island to another in the cluster. Minimum Cost spanning tree has a broad application in various fields. Minimum spanning tree can solve many complicated problems in the real world, such as:

- Minimum distance for traveling to all cities at one time (Traveling salesman problem)
- In electronic circuit design, to connect cable efficiently
- Stretched trees also seek their application in obtaining a free set of circuits from an electrical network.

2.2. Prim's Algorithm

The prim algorithm is a greedy method that helps us get a minimum stretched tree. The prim algorithm uses the set concept. Processing graphs by sorting sides. This algorithm processes the sides of the Graph randomly by constructing disjoint from sets. The prim algorithm constructs a minimum spanning tree through the sequence of subtree propagation. The first subtree of a sequence consists of a single vertex selected from set V of the graph vertices. In each iteration, this algorithm unfolds a tree by using the greedy technique by linking it to the nearest node that is not in the tree created (with the closest node means that the node is not connected to the existing node in the tree with the minimum weighted side). This algorithm will stop until all Graf nodes have been entered into the tree that is created and connected without forming a circuit.

There are two disjoint sets of vertices. One of the disjoint sets contains vertices in growing spanning tree, and the other contains vertices that are not in growing spanning tree. Choose the lightest vertex that is connected by growing spanning tree and not in growing spanning tree then add it to growing spanning tree. It can be done using a Priority Queue. Insert a vertex in the growing spanning tree into Priority Queue. Make sure no cycle occurs. To do this, mark the selected node and only insert nodes that are not marked into the Priority Queue [15].

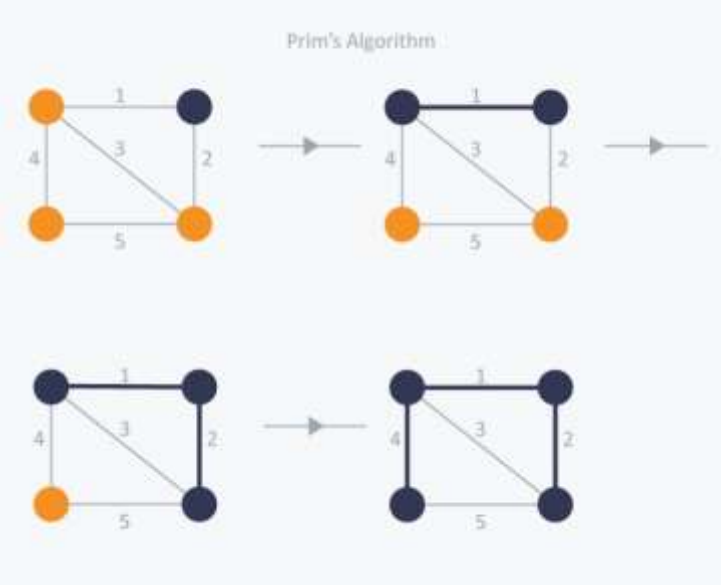

Figure 1 Way the Prim algorithm works

Figure 1 describes the workflow of the Prims algorithm. The journey starts from the arbitrage node and is marked. In each iteration, new vertices adjacent to the marked vertices

will be marked. Because this algorithm uses a greedy approach, the Prim algorithm will select the smallest edge and then mark the vertex. In other words, the edge with a weight of 1 will be selected first. In the next iteration, there are three options; edge with weights 2, 3, and 4. Edge with a weight of 2 will be selected, and the corresponding vertex will be marked. Now, there are three options; edge with weights 3, 4 and 5. Weight 3 cannot be chosen because it will lead to a cycle. Therefore, the edge with weight 4 will be selected, and the result of the minimum spanning tree is 1 + 2 + 4 = 7.

2.3. Genetic Algorithm

Genetic Algorithms are heuristic search algorithms based on the mechanism of biological evolution. Diversity in biological evolution is the variation of chromosomes between individual organisms [16]–[18] This chromosome variation will affect the rate of reproduction and the level of the organism's ability to stay alive. The Genetic Algorithm uses the principle of finding neighborhood solutions based on natural selection and natural genetic mechanisms that can be used to solve optimization problems such as the Minimum Spanning Tree problem [19]–[21].

In the Genetic Algorithm, solving the Minimum Spanning Tree problem is almost the same as the solution to other optimization problems. The only difference is the chromosome coding process (encoding), weight calculation (decoding) and recombination or crossover. In addition to the case of the Minimum Spanning Tree, another step is added, namely the process of modifying the degree to check the presence of a circuit or not. The steps of the Genetic Algorithm in the Minimum Spanning Tree are as follows:

Chromosome Encoding (Encoding)

In the chromosome coding (Algorithm) genetic algorithm it is essential to represent the initial problem. In the Minimum Spanning Tree (MST), chromosomes are representations of Spanning Tree and Gen representations of the nodes that make up the tree. Chromosome coding in MST uses a chromosomal decoder technique called Prufer Number. The Prufer Number encoding aims to encode the trees generated from a graph so that Genetic Algorithms can solve the MST problem. The Prufer Number steps are as follows:

Step 1: Pay attention to all points 1 and select the smallest point

Step 2: Pay attention to the points adjacent to the point we have chosen in step 1, and place this label in the first position of the line that will be formed

Step 3: Remove the point selected in step 1 along with the insulating side, until the smaller tree remains. Repeat steps 1-3 for the remaining trees. Continue up to two points. When it is finished, the desired Prufer Number row has been formed, where the node code number is n-2 (n number of vertices of the graph).

Weight Calculation (Decoding)

The decoding process is useful for encoding the values of individual forming genes or turning each chromosome into a tree, so that the number of weights of each tree can be calculated. The steps for decoding Prufer Numbers on MST are:

- Suppose that P is the Prufer Number of the tree and Q is the set of vertices that are not in P. Q is called the eligible vertex
- Suppose i is the leftmost digit of P and j is a node of Q where j is the node with the smallest number. Add sections from i to j. Discard j from Q. If i no longer exist in P, place j on Q. Repeat this process so that there are no more P digits left.

- Add the last 2 remaining segments so that eventually form a tree with n-1 segment. If the decoding process has been completed, then the fitness value can be calculated and arises the initial isolation and selection of the genetic algorithm.

Recombination or Crossing (Crossover)

Crossover (crossing) is performed on two chromosomes to produce a child chromosome (offspring). The crossover method that is most commonly used in genetic algorithms is the one-point crossover method. Crossover steps are:

- Revive random Rk values as much as the existing population.
- The number of chromosomes to be crossed is as much as the crossover (Pc) opportunity specified. Crossover opportunities are determined by the user
- Determine the position of the intersection by generating a random number 1 to the length of the chromosome-1 (if the length of chromosome 5, the position of the intersection point 1-4).
- Exchange the right side of the cut point from both parts of the parent chromosome
- Furthermore, chromosomes can be mutated

Mutation

The mutation used in the genetic algorithm will randomly change the value of a bit in a certain position. In this mutation, it is possible to emerge a new chromosome which originally did not appear inside the initial population. The mutation step is:

- Generate random values between 0-1
- Determine Mutation Opportunities (Pm)
- By using the chance of a Pm mutation, it is expected that as much as the chance of mutation of the total gene will mutate.
- Chromosomes with random values < Pm will be mutated, by switching genes according to mutation positions. The position of the mutation is determined by generating a random number 1 to the length of the chromosome
- The final population of this mutation process will be used as the initial population for the next generation.

Chromosome preservation

Because the selection technique in the genetic algorithm is done randomly, so there is a possibility that a good chromosome cannot enter the next generation because it is not selected in the selection, it is necessary to preserve chromosome. Chromosome preservation steps are:

- Determine the chances of chromosome sustainability
- Generate as many random numbers as the population. Chromosomes with random values < chance of survival will be affected by chromosomal changes.

3. RESULT AND DISCUSSION

This part is the design of graph formation. Some graph models will be created to test the ability of both algorithms in determining minimum spanning tree. The following table is the graph data used in this research. Each node has different coordinate location.

Table 1 Location of X-axis and Y-axis

Location	X	Y
L1	38	29
L2	46	76
L3	90	108
L4	150	40
L5	107	73

Each node has a certain distance from another node. The distance calculation is obtained by using Euclidean Distance. The following illustration is the distance of each node.

L1-L2 = $\sqrt{(38-46)^2+(29-79)^2}$

= $\sqrt{64+2209}$

= $\sqrt{2273}$

= 47.68

L1-L3 = $\sqrt{(38-90)^2+(29-108)^2}$

= $\sqrt{2704+6241}$

= $\sqrt{8945}$

= 94.58

L1-L4 = $\sqrt{(38-150)^2+(29-40)^2}$

= $\sqrt{12544+121}$

= $\sqrt{12665}$

= 112.54

L1-L5 = $\sqrt{(38-107)^2+(29-73)^2}$

= $\sqrt{4761+1936}$

= $\sqrt{6697}$

= 81.84

L2-L3 = $\sqrt{(46-90)^2+(76-108)^2}$

= $\sqrt{1936+1024}$

= $\sqrt{2960}$

= 54.41

L2-L4 = $\sqrt{(46-150)^2+(76-40)^2}$

= $\sqrt{10816+1296}$

= $\sqrt{12112}$

= 110.05

L2-L5 = $\sqrt{(46-107)^2+(76-73)^2}$

= $\sqrt{3721+9}$

= $\sqrt{3730}$

= 61.07

L3-L4 = $\sqrt{(90-150)^2+(108-40)^2}$

= $\sqrt{3600+4624}$

= $\sqrt{8224}$

= 90.69

L3-L5 = $\sqrt{(90-107)^2+(108-73)^2}$

= $\sqrt{289+1225}$

= $\sqrt{1514}$

= 38.91

L4-L5 = $\sqrt{(150-107)^2+(40-73)^2}$

= $\sqrt{1849+1089}$

= $\sqrt{2938}$

= 54.2

Table 2 Location using Euclidean Distance

	L1	**L2**	**L3**	**L4**	**L5**
L1	0	47.68	94.58	112.54	81.84
L2	47.68	0	54.41	110.05	61.07
L3	94.58	54.41	0	90.69	38.91
L4	112.54	110.05	90.69	0	54.2
L5	81.84	61.07	38.91	54.2	0

Table 2 describes distances between locations (L1-L5). A zero value means there is no distance traveled or the location does not move to another location.

3.1. Genetic Algorithm Test

Table 3 Inital Random Population

4	1	0	2	3
2	0	3	4	1
2	3	1	0	4
4	2	0	1	3
1	3	0	2	4
4	2	1	0	3
1	0	3	4	2
3	2	4	1	0
4	2	1	0	3
1	2	0	3	4

Genetic algorithm testing in the case of the Minimum Spanning Tree involves six genes. This test involves 10 populations and 20 generations. The population is created randomly with a certain number. The value of each population may not appear more than one. The following tables are the the genetic algorithm process.

The parameter used are "Generation = 20" and "Population = 10". The following illustration is the process carried out by the Genetic algorithm.

Table 4 Last Generation

3	1	0	2	4
2	1	0	4	3
4	3	2	0	1
3	0	4	1	2
3	0	2	1	4
3	4	1	0	2
4	3	2	0	1
0	1	4	3	2
4	2	0	3	1
3	0	4	1	2

Table 3 Fitness Calculation

Population	**Weight**	**Fitness**
1	203	0.005050505
2	216	0.005405405
3	277	0.008064516
4	225	0.005681818
5	210	0.005235602
6	211	0.005263158
7	277	0.008064516
8	302	0.01010101
9	282	0.008403361
10	225	0.005681818

Table 4 Population After Selection

3	1	0	2	4
4	3	2	0	1
3	0	4	1	2
2	1	0	4	3
4	2	0	3	1
3	4	1	0	2
0	1	4	3	2
4	2	0	3	1
4	3	2	0	1
4	3	2	0	1

Table 5 Population After Mutation

3	1	0	2	4
0	3	2	4	1
3	2	0	1	4
3	1	2	4	0
4	3	2	1	0
0	1	4	3	2
0	4	1	2	3
4	3	2	1	0
3	2	1	0	4
3	2	1	0	4

Table 6. Fitness After Elitism

Population	Weight	Fitness
1	203	0,005050505
2	304	0,010309278
3	207	0,005154639
4	222	0,005586592
5	292	0,009174312
6	302	0,01010101
7	226	0,005714286
8	292	0,009174312
9	212	0,005291005
10	212	0,005291005

Table 7 Population After Elitism

3	1	0	2	4
0	3	2	4	1
3	2	0	1	4
3	1	2	4	0
4	3	2	1	0
0	1	4	3	2
0	4	1	2	3
4	3	2	1	0
3	2	1	0	4
3	2	1	0	4

The last generation shows the most optimum results for the whole generation and population. It is visible from the displayed list. Population 1 is the best value of the ten populations. The route obtained is 3-1-0-2-4 (L4-L2-L1-L3-L5) which has a distance of 203.

3.2. Prim Algorithm Test

Prim looks for the next destination with the smallest distance. If there are several nodes as an alternative, then the selected node is the shortest node from the beginning. Calculations are applied to all five locations (L1-L5). The first location (L1) has four possible goals (L2-L5), L2 has three other goals, L3 has two objectives, and L4 has one goal. The following table is the result of the Prim algorithm calculation.

Table 3 Prim algorithm result

L1	to	L2	47.68	47.68	L2
		L3	94.58		
		L4	112.54		
		L5	81.84		
L2	to	L3	54.41	54.41	L3
		L4	110.05		
		L5	61.07		
L3	to	L4	90.69	38.91	L5
		L5	38.91		
L5	to	L4	54.2	54.2	L4

The route of the Minimum Spanning Tree generated by Prim's algorithm is L1-L2-L3-L5-L4 with weights 47.68 + 54.41 + 38.91 + 54.2 = 195.2. The Prim algorithm is better than Genetics in both tests. The difference is seen as 203 - 195.2 = 7.8.

4. CONCLUSIONS

The prim algorithm operates at two disjoint from the side of the Graph. The prim algorithm can solve a problem with a better time if there are fewer vertices and edges. This prim algorithm is just one of the algorithms to solve the problem of minimum cost spanning tree. The test results state that the Prim algorithm has better speed than the Genetic algorithm. It is proven by the accuracy of Prim's algorithm which is better than Genetics. In significant cases, Genetic algorithms are better at finding optimal routes. It is because Genetic algorithms have the probability to provide the possibility of opportunities to get the next generation.

REFERENCES

[1] S. Aryza, M. Irwanto, Z. Lubis, A. P. U. Siahaan, R. Rahim, dan M. Furqan, "A Novelty Design of Minimization of Electrical Losses in A Vector Controlled Induction Machine Drive," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 300, no. 1.

[2] Khairul *et al.*, "Effect of Matrix Size in Affecting Noise Reduction Level of Filtering," *Int. J. Eng. Technol.*, vol. 7, no. 3, hal. 1272–1275, 2018.

[3] A. P. U. Siahaan *et al.*, "Combination of Levenshtein Distance and Rabin-Karp to Improve the Accuracy of Document Equivalence Level," *Int. J. Eng. Technol.*, vol. 7, no. 2.27, hal. 17–21, 2018.

[4] A. Ikhwan, M. Yetri, Y. Syahra, dan J. Halim, "A Novelty of Data Mining for Promoting Education based on FP-Growth Algorithm," *Int. J. Civ. Eng. Technol.*, vol. 9, no. 7, hal. 1660–1669, 2018.

[5] Andre Hasudungan Lubis, S. Z. S. Idrus, dan A. Sarji, "ICT Usage Amongst Lecturers and Its Impact Towards Learning Process Quality," *Malaysian J. Commun.*, vol. 34, no. 1, hal. 284–299, 2018.

[6] R. Rahim *et al.*, "Combination Base64 Algorithm and EOF Technique for Steganography," *J. Phys. Conf. Ser.*, vol. 1007, no. 1, hal. 1–5, 2018.

[7] R. Meiyanti, A. Subandi, N. Fuqara, M. A. Budiman, dan A. P. U. Siahaan, "The Recognition of Female Voice Based on Voice Registers in Singing Techniques in Real-

Time using Hankel Transform Method and Macdonald Function," *J. Phys. Conf. Ser.*, vol. 978, no. 1, hal. 1–6, 2018.

[8] A. P. U. Siahaan dan R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm," *Int. J. Secur. Its Appl.*, vol. 10, no. 8, hal. 173–180, Agu 2016.

[9] M. Townsend, *Discrete Mathematic: Applied Combinatorics and Graph Theory*. The Benjamin/Cummings Publishing Company, 1987.

[10] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent System*. England: Addison-Wesley, 2005.

[11] A. P. U. Siahaan, "Heuristic Function Influence to the Global Optimum Value in Shortest Path Problem," *IOSR J. Comput. Eng.*, vol. 18, no. 05, hal. 39–48, Mei 2016.

[12] D. Kalpanadevi, "Effective Searching Shortest Path In Graph Using Prim's Algorithm," *Int. J. Comput. Organ. Trends*, vol. 3, no. 8, hal. 310–313, 2013.

[13] M. Iqbal, A. P. U. Siahaan, N. E. Purba, dan D. Purwanto, "Prim's Algorithm for Optimizing Fiber Optic Trajectory Planning," *Int. J. Sci. Res. Sci. Technol.*, vol. 3, no. 6, hal. 504–509, 2017.

[14] Z. Ramadhan, "Perbandingan Algoritma Prim dan Algoritma Floyd-Warshall dalam Menentukan Lintasan Terpendek (Shortest Path Problem)," Universitas Sumatera Utara, 2016.

[15] Jawahar, "Minimum Spanning Tree (Pohon Rentangan Minimum)," 2017. [Daring]. Tersedia pada: https://indonesia.hackerearth.com/minimum-spanning-tree-pohon-rentangan-minimum/. [Diakses: 01-Okt-2018].

[16] S. Gupta dan P. Panwar, "Solving Travelling Salesman Problem Using Genetic Algorithm," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 6, hal. 376–380, 2013.

[17] H. Tabatabaee, "Solving the Traveling Salesman Problem using Genetic Algorithms with the New Evaluation Function," *Bull. Environ. Pharmacol. Life Sci.*, vol. 4, no. 11, hal. 124–131, 2015.

[18] Z. Ramadhan, A. Putera Utama Siahaan, dan M. Mesran, "Prim and Floyd-Warshall Comparative Algorithms in Shortest Path Problem," in *Proceedings of the Joint Workshop KO2PI and The 1st International Conference on Advance & Scientific Innovation*, 2018.

[19] A. P. U. Siahaan, "Genetic Algorithm in Hill Cipher Encryption," *Am. Int. J. Res. Sci. Technol. Eng. Math.*, vol. 15, no. 1, hal. 84–89, 2016.

[20] A. Philip, A. A. Taofiki, dan O. Kehinde, "A Genetic Algorithm for Solving Travelling Salesman Problem," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 1, hal. 26–29, 2011.

[21] Y. F. Waruwu, "Analisis Nilai Mutasi Dinamis pada Algoritma Genetika," Universitas Sumatra Utara, 2016.