International Journal of Information System & Technology
Akreditasi No. 158/E/KPT/2021 | Vol. 5, No. 5, (2022), pp. 541-547

Utilization of The Generate and Test Algorithm In Shortest Route Search Case

Silvilestari

Informatics Management, Akademi Manajemen Informatika Dan Komputer Kosgoro, Solok, West Sumatra, Indonesia

rendanghjfatimah@gmail.com

Abstract

The Gerenete and test algorithm is a very simple method in a heuristic search. The principle of this method is a combination of the depthbforst search method with backtracking because it produces a complete solution before the test process is carried out. If the work process is carried out systematically and follows the procedure, it will find a solution. The problem in this research is that if the problem is broad and large, it will likely take time to find a solution. The main purpose of this research is to test the distance in order to find a solution so as to produce a solution that is fast and saves time. Generate and test work system, namely 1. Generate a possible solution 2. Test each node which is a solution by comparing the node with the end node of a selected path with a set of expected goals. 3. If a solution has been found, then exit the system. If you haven't found a solution, then return to step 1. The final result of the research is the point F(A-B-E-I-L-T) with a heuristic number of 14.

Keywords: *Generete Algorithm and test;Heuristics; Grap, Distance between Nodes*

1. Introduction

Some of the graph terminology discussed are vertices, edges, undirected graphs, directed graphs, weighted graphs, and paths. Vertex is a point or node that shows a place that is used as a goal or destination, start or beginning, or a place to be traversed in a journey. Edge is a connecting line between vertices. An undirected graph is a graph that has no direction and arrows so that it can be traversed by two opposite directions. A directed graph is a graph that has a direction and is usually indicated by an arrow with one end called the tail and head. A directed graph cannot be traversed in two different directions. A weighted graph is a graph that has a weight or value at each edge and can be either a directed graph or an undirected graph. Path is a path that passes through vertices and edges where vertices cannot be passed more than once [1]. Several heuristic search concepts. According to Kusumadewi (2003), in principle this method is a combination of depth-first search with backtracking, namely moving backwards towards an initial state. The test value is in the form of a 'yes' or 'no' answer. The algorithm of the Generate and Test method is:

a) Generate a possible solution (generate a certain point or a certain path from the initial state).
b) Test to see if the node is really the solution by comparing the node or the end node of a selected path with the expected set of destinations.
c) If a solution is found, exit. If not, repeat the first step [2].

According to research [3] To overcome this, a system is needed that can provide information on the presence of the nearest gas station. Previously, research has been carried out in the form of searching for the nearest gas station and determining the shortest distance using the Djikstra algorithm [1]. Research uses location-based services based on sensors on gasoline indicators for Android phones in finding the nearest gas station [2]. Other research uses Quantum Geographic Information Systems (QGIS) to get

ISSN : 2580-7250

the location of the nearest gas station and identify the closest path from users to the gas station [3]. However, from the three studies, it is not possible to know whether the gas station is still operating and fuel oil is still available.

Research that has been researched related to Generete and Test includes several studies related to the use of generate and test methods including, [1] Analysis Performance of Generate And Test Algorithm in Eye Disease Diagnosis Expert System, [2] Predicting The Selling Price Of Dried Eucheumacottoni in Indonesia With Four Classifier Of Data Mining Techiniques, [3] Minimizing Forecasting Error Values With Extreme Learning Machine Algorithms, [4] Implementation Generate And Test Method in Solving Traveling Salesman Problem Using Sonar and Color Sensor Robot, [5] Implementation of Generate And Test Algorithm in Searching the Shortest Route.

When considering intelligent information processing systems operating in complex environments under various time constraints, as defined by Wang (2006), the need for introspective capabilities on the part of the system quickly becomes apparent. Proficiency under limited knowledge and resources is central to this view of intelligence. In an environment that produces abundant sensory information resource management is a particularly important issue because the processing capacity of the system is greatly controlled by the amount of information generated by the environment, which requires the system to continuously monitor and anticipate resource usage - an activity that again requires introspective functionality. Meta-learning, self-configuration and adaptive attention are other examples of introspective operations, some - and possibly all - that are possible [6]. The history of TSP starts from Euler who studied the Knight Tour's Problem (1759), Kirkman who studied polyhedron graphs (1856) and Hamilton who made Icosian (1856) which aimed to find a circuit path based on a polyhedron graph that satisfies certain conditions [2]. The term CSR itself is thought to have come from a book published by a veteran salesman in the 1930s in Germany, although in this book the issue of CSR is discussed more from the business aspect and has not been formulated mathematically. The Traveling Salesman Problem was raised in 1800 by the Irish mathematician William Rowan Hamilton and the British mathematician Thomas Penyngton. from the Icosian Hamilton game which requires the player to complete a journey of 20 points using only certain paths. The general form of TSP was first studied by mathematicians starting in 1930. It was started by Karl Menger in Vienna and Harvard. After that the TSP problem was published by Hassler Whitney and Merrill Flood at Princeton. Furthermore, with this problem, CSR has become a well-known and popular problem to be used as a production, transportation and communication model [4].

Several methods can solve TSP, including: Linear Programming (LP), Genetic Algorithm, Nearest Neighborhood Heuristic (NNH) and Cheapest Insertion Heuristic (CIH). According to Sutojo, et al. (2011), the heuristic search technique is a strategy to conduct the search process selectively and can guide the search process that has the greatest probability of success. Heuristic Algorithm is one of the alternative algorithms that can be used for this problem, because the process is fast and gives the desired results. Commonly used algorithms in heuristic searches are Generate and Test, Hil Caimbing, A*, Best First Search, and Recursive Best First Search [4]. The combination of the advantages of the Depth First and Breath First Search methods. The combination of these two search techniques is used to browse one path at a time, but can switch when the other path looks better than the path being searched. In each state generated with the heuristic function, a priority scale is given to obtain the best path [3].

2. Research Methodology

Research methodology is the process of doing research from the beginning of research to producing value. To obtain good research values, you must use a systematic system. The process of generate and test research methodology is:

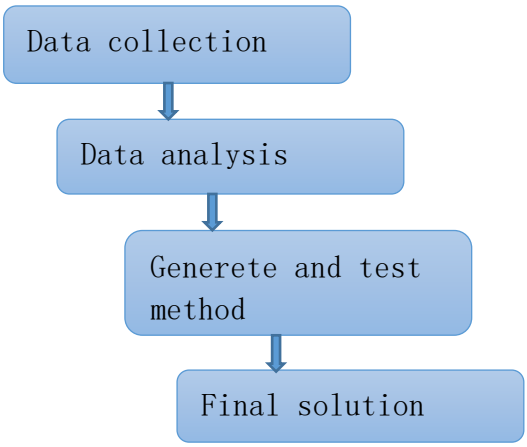

Figure 1. Research methodology

3. Results and Discussion

Implementation of the General Algorithm and Test In determining the shortest route can be seen in the image below:

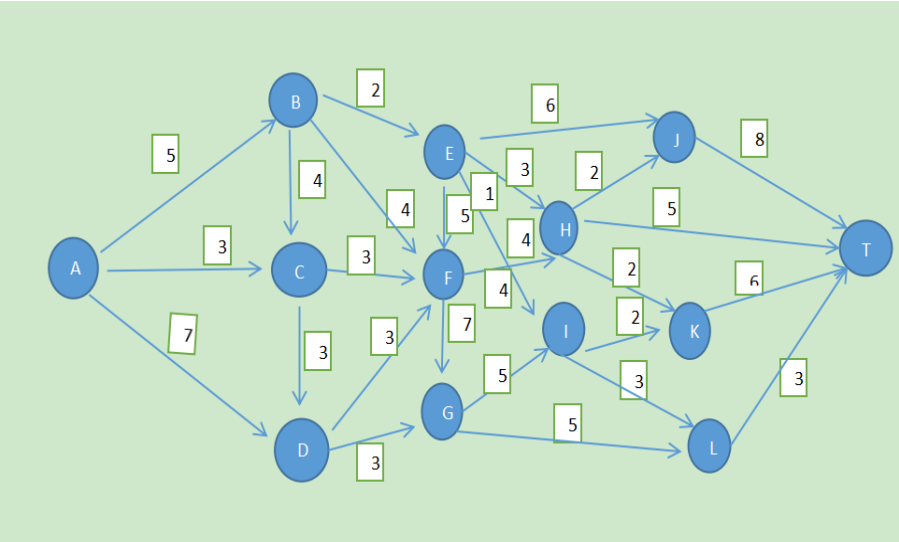

Figure 2. Data analysis

In the picture above, it can be calculated that the number of cities to be visited is 13 cities, namely City A, B, C, D, F, G, H, I, J, K, L, the distance between each city can be seen in the image above. The arrow attribute used is part of the sign of the path to be traversed. The problem in this case is that a daily goods salesman is currently in City A, the salaemen wants to deliver goods to city T, which route must be taken by the daily goods salesman to arrive at the city of Zagar has a fast path and has efficient time and distance. at least.

From the case of the daily goods salesman, it can be seen that:

Initial State = A

Goal = T

The solution that will be done is in the form of the first path that is passed by path A and the last path is path T. To solve general and test problems is the Deep First Search algorithm. Efficient acquisition of time and distance requires the shortest distance. Because what will be sought is in the form of the shortest path, the criteria used are in the form of finding the shortest route solution, in this case a distance heuristic function is

needed as a clue to get the results. Below is an example of a solution search iteration 1-10 as a sample for easy understanding.

General Algorithm Process and Test

a) Iteration to -1

Generate all existing algorithms using solutions with systems using the initial DFS algorithm including:

A-B-E-J-T =F(A- B-E-J-T)= 5+2+6+8 =21

The iteration test results used were 21, so it was used as a temporary solution, namely = F(A- B-E-J-T).

b) Iteration -2

Generate all existing algorithms using a solution with the system using the second DFS algorithm including:

A-B-E-I-K-T= F(A-B-E-I-K-T)=20

The iteration test results used as many as 20, then it is used as a temporary solution, namely F= (A-B-E-I-K-T).

c) Iteration -3

Generate all existing algorithms using a solution with a system using the Third DFS algorithm including:

A-B-E-I-L-T= F(A-B-E-I-L-T)=14

The iteration test results used as many as 14, then used as a temporary solution, namely F = (A-B-E-I-L-T).

d) Iteration -4

Generate all existing algorithms using solutions with systems using the Fourth DFS algorithm including:

A-B-E-I-L-T F(A-B-E-H-T)=15

The iteration test results used as many as 14, then used as a temporary solution, namely F= (A-B-E-I-L-T).

e) Iteration to -5

Generate all existing algorithms using solutions with systems using the Fifth DFS algorithm including:

F(A-B-E-I-L-T)=14

The iteration test results used as many as 14, then used as a temporary solution, namely F= (A-B-E-I-L-T).

f) Iteration -6

Generate all existing algorithms using solutions with systems using the six DFS algorithms including:

F(A-C-F-H-J-T)=20

The iteration test results used as many as 14, then used as a temporary solution, namely F= (A-B-E-I-L-T).

g) Iteration -7

Generate all existing algorithms using solutions with systems using the Seventh DFS algorithm including:

F(A-C-F-H-T)=15

The iteration test results used as many as 14, then used as a temporary solution, namely F = (A-B-E-I-L-T).

h) Iteration to -8

Generate all existing algorithms using solutions with systems using the Eighth DFS algorithm including:

A-C-F-H-K-T=18

The iteration test results used as many as 14, then used as a temporary solution, namely F= (A-B-E-I-L-T).

i) Iteration -9

Generate all existing algorithms using solutions with systems using the Ninth DFS algorithm including:
A-D-F-H-J-T
The iteration test results used as many as 14, then used as a temporary solution, namely F= (A-B-E-I-L-T).

j) Iteration to -10

Generate all existing algorithms using solutions with systems using the Tenth DFS algorithm including:
F(A-D-G-L-T)=18
The iteration test results used as many as 14, then used as a temporary solution, namely F= (A-B-E-I-L-T).

Manual calculation process

1) Iteration 1 = A-B-E-J-T = 5+2+6+8=21
2) Iteration 2 = A-B-E-H-T= 5+2+3+5=15
3) Iteration 3 = A-B-E-F-H-J-T= 5+2+5+2+8=22
4) Iteration 4 = A-B-E-H-K-T= 5=2+3+2+6=18
5) Iteration 5= A-B-E-I-K-T= 5+2+5+2+6=20
6) Iteration 6= A-B-E-I-L-T =5+2+1+3+3=14
7) Iteration 7= A-B-E-F-G-I-K-T= 5+2+5+7+5+2+6=40
8) Iteration 8= A-B-E-F-G-L-T= 5+2+5+7+5+3=27
9) Iteration 9= A-B-C-F-H-J-T=5+4+3+4+2+8=26
10) Iteration 10= A-B-C-F-H-K-T=5+4+3+4+2+6=24
11) Iteration 11= A-B-C-F-G-I-K-T= 5+4+3+5+2+6=25
12) Iteration 12= A-B-C-F-G-L-T= 5+4+3+7+5+3=27
13) Iteration 13= A-B-C-D=F-H-J-T = 5+4+3+3+4+2+8=29
14) Iteration 14= A-B-C-D-G-I-K-T = 5+4+3+3+5+2+6=28
15) Iteration 15= A-B-C-D-G-L =5+4+3+3+5=20
16) Iteration 16= A-B-C-D-F-H-J-T= 5+4+3+4+2+8=26
17) Iteration 17= A-B-C-D-I-K-T= 5+3+4+3+5+2+6=28
18) Iteration 18= A-C-F-H-J-T= 3+3+4+2+8=20
19) Iteration 19= A-C-F-H-T= 3+3+4+5=15
20) Iteration 20= A-C-F-H-K-T= 3+3+4+4+2+8=24
21) Iteration 21= A-C-D-F-H-J-T= 3+3+3+4+2+8=23
22) Iteration 22= A-C-F-H-K-T=3+3+2+6=14
23) Iteration 23= A-C-D-F-H-J-T=3+3+3+4+5=23
24) Iteration 24= A-C-D-F-H-T= 3+3+3+4+5=18
25) Iteration 25= A-C-D-G-I-K-T= 3+3+3+2+5+6=21
26) Iteration 26 A-C-D-G-I-L-K= 3+3+3+5+3+3=20
27) Iteration 27= A-D-F-H-J-T=7+3+4+2+8=24
28) Iteration 28= A-D-G-L-T=3+3+5+3=14
29) Iteration 29= A-D-G-I-K-T=3+3+5+2+6=17

Table of accumulation of all search processes that have found solutions so that they can be sorted according to the values obtained.

Table 1. Process Table Generate and test

Iteration Number	City name	The final result	Selected track
1	A-B-E-J-T	5+2+6+8=21	21
2	A-B-E-H-T	5+2+3+5=15	15
3	A-B-E-F-H-J-	5+2+5+2+8=22	15

Iteration Number	City name	The final result	Selected track
4	A-B-E-H-K-T	5=2+3+2+6=18	15
5	A-B-E-I-K-	5+2+5+2+6=20	15
6	A-B-E-I-L-T	5+2+1+3+3=14	14
7	A-B-E-F-G-I-K-T	5+2+5+7+5+2+6=40	14
8	A-B-E-F-G-L-T	5+2+5+7+5+3=27	14
9	A-B-C-F-H-J-	5+4+3+4+2+8=26	14
10	A-B-C-F-H-K-T	5+4+3+4+2+6=24	14
11	A-B-C-F-G-I-K-T=	5+4+3+5+2+6=25	14
12	A-B-C-F-G-L-T=	5+4+3+7+5+3=27	14
13	A-B-C-D=F-H-J-	5+4+3+3+4+2+8=29	14
14	A-B-C-D-G-I-K-T	5+4+3+3+5+2+6=28	14
15	A-B-C-D-G-L	=5+4+3+3+5=20	14
16	A-B-C-D-F-H-J-T=	5+4+3+4+2+8=26	14
17	A-B-C-D-I-K-T=	5+3+4+3+5+2+6=28	14
18	A-C-F-H-J-T=	3+3+4+2+8=20	14
19	A-C-F-H-T=	3+3+4+5=15	14
20	A-C-F-H-K-T=	3+3+4+4+2+8=24	14
21	A-C-D-F-H-J-T=	3+3+3+4+2+8=23	14
22	A-C-F-H-K-T=	3+3+2+6=14	14
23	A-C-D-F-H-J-T=	3+3+3+4+5=23	14
24	A-C-D-F-H-T=	3+3+3+4+5=18	14
25	A-C-D-G-I-K-T=	3+3+3+2+5+6=21	14
26	A-C-D-G-I-L-K=	3+3+3+5+3+3=20	14
27	A-D-F-H-J-T	=7+3+4+2+8=24	14
28	A-D-G-L-T	=3+3+5+3=14	14
28	A-D-G-I-K-T	=3+3+5+2+6=17	14

Based on the final results of the entire search process, 3 point values are obtained that have the same distance results, namely:

a. A-B-E-I-L-T = 14
   = 5+2+1+3+3=14
b. A-C-F-H-K-T = 14
   = 3+3+2+6=14
c. A-D-G-L-T = 14
   = 3+3+5+3=14

To see the final results of the search process can be seen in the image below:

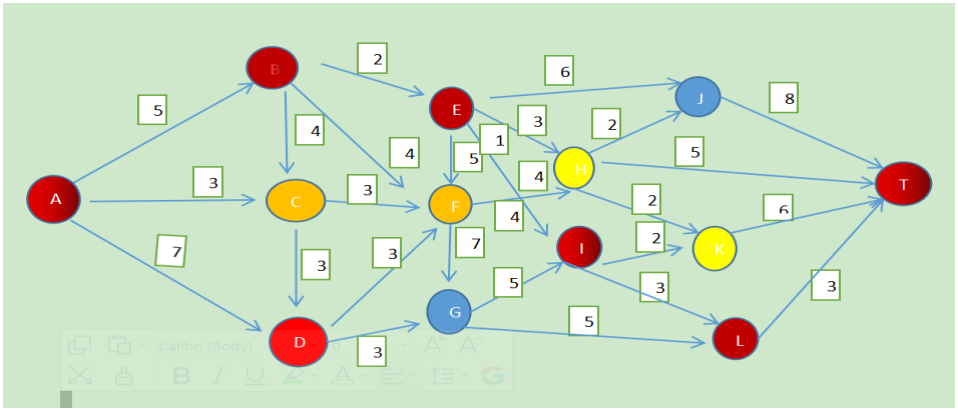

Figure 3. General And Test Algorithm Final Results

The final result of the 13 routes that will be passed by the salesmen, the closest count value is obtained as many as 14 distance values that are the least traveled by the salesmen. To obtain the maximum value, a systematic calculation method is needed to obtain the final solution.

4. Conclusion

Based on the testing and analysis of the system and taking into account the entire process that occurs, the Generate and Test algorithm is able to find the shortest route in solving the settlement case. With the Generate and Test Algorithm able to find the shortest route in solving settlement cases. So that it can be used as an alternative to find a solution in order to save time visiting the point to be visited. The results of this study found 3 alternatives that can be used as references by salesmen to visit the point to be addressed. The smallest route obtained a value of 14 where the route consists of A-B-E-I-L-T, A-C-F-H-K-T and A-D-G-L-T. The Generate and Test method can be used as a reference in determining the shortest route to produce a solution.

References

[1] Angriani, H., & Saharaeni, Y. (2020). Implementasi Algoritma Best First Search Dalam Sistem Pakar Pertolongan Pertama Pada Bayi dan Anak. *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, *10*(2), 116. https://doi.org/10.35585/inspir.v10i2.2575

[2] Apriandi, R., Rismawan, T., & Midyanti, D. M. (2018). *Jurnal Coding , Sistem Komputer Untan ISSN : 2338-493X Penerapan Metode Best First Search (Bfs) Untuk Pencarian Lokasi Spbu Terdekat Menggunakan Jurnal Coding , Sistem Komputer Untan ISSN : 2338-493X*. *06*(1), 1–11.

[3] Dangkua, E. V., Gunawan, V., & Adi, K. (2015). Penerapan Metode Hill Climbing Pada Sistem Informasi Geografis Untuk Mencari Lintasan Terpendek. *Jurnal Sistem Informasi Bisnis*, *5*(1), 19–25. https://doi.org/10.21456/vol5iss1pp19-25

[4] Putri, N. I., & Munawar, Z. (2019). Mekanisme umum untuk sistem kecerdasan buatan. *COMPUTING| Jurnal Informatika*, *06*(02), 58–75. http://ejournal.unibba.ac.id/index.php/computing/article/view/206

[5] Rahmadi, R. (2010). Implementasi Metode Generate and Test Dalam Menyelesaikan Travelling Salesman Problem Menggunakan Robot Bersensor. *Seminar Nasional (SNATI)*, *2010*(Snati), 29–34.

Welianto, S., Santosa, R. G., & Chrismanto, A. R. (2012). Implementasi Algoritma Generate and Test Pada Pencarian Rute Terpendek. *Jurnal Informatika*, *7*(2), 1–10. https://doi.org/10.21460/inf.2011.72.103