International Journal of Advances in Data and Information Systems
Vol. 2, No. 1, April 2021, pp. 45~52
ISSN: 2721-3056, DOI: 10.25008/ijadis.v2i1.1206

Greedy, A-Star, and Dijkstra's Algorithms in Finding Shortest Path

M. Rhifky Wayahdi[1], Subhan Hafiz Nanda Ginting[2], Dinur Syahputra[3]
[1,2,3]Department of Information System, Faculty of Technology, Battuta University, Indonesia
[1]muhammadrhifkywayahdi@gmail.com, [2]subhanhafiz16@gmail.com, [3]dinsyahui12@gmail.com

Article Info

Article history:

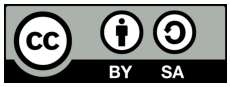
Received Dec 19, 2020
Revised Jan 16, 2021
Accepted Feb 01, 2021

Keywords:

Greedy algorithm
A-Star algorithm
Dijkstra's algorithm
Path finding
Shortest path

ABSTRACT

The problem of finding the shortest path from a path or graph has been quite widely discussed. There are also many algorithms that are the solution to this problem. The purpose of this study is to analyze the Greedy, A-Star, and Dijkstra algorithms in the process of finding the shortest path. The author wants to compare the effectiveness of the three algorithms in the process of finding the shortest path in a path or graph. From the results of the research conducted, the author can conclude that the Greedy, A-Star, and Dijkstra algorithms can be a solution in determining the shortest path in a path or graph with different results. The Greedy algorithm is fast in finding solutions but tends not to find the optimal solution. While the A-Star algorithm tends to be better than the Greedy algorithm, but the path or graph must have complex data. Meanwhile, Dijkstra's algorithm in this case is better than the other two algorithms because it always gets optimal results.



Corresponding Author:

M. Rhifky Wayahdi,
Department of Information System,
Faculty of Technology
Battuta University,
Gadjah Mada Road, Number 15 M, Medan, North Sumatra, Indonesia.
Email: muhammadrhifkywayahdi@gmail.com

1. INTRODUCTION

Finding the shortest path in a graph is a problem that is often discussed and has been widely applied in optimizing the running of a system process. Edges and Vertices can be used to solve the problem of finding the shortest distance. There are several algorithms or methods that can be used in the process of finding a path with a minimum distance between the source to the destination such as the Greedy Algorithm, A-Star, Dijkstra, and so on.

Bhosale *et al.* in his research states that the Greedy algorithm can be close to the optimum to approach the factor (*1-1e*). However, this algorithm is quite expensive to overcome the problem maximization effect on large scale networks [1]. Zhenhuan in his research explained, the greedy algorithm can reach a good solution in a short time, the experimental data shows that it is quite effective [2]. Meanwhile, Patil & Amaratir concluded that the greedy algorithm is faster and more efficient in terms of the number of iterations needed to segment objects from an image compared to the M Kass algorithm [3].

Duchon *et al.* in his research stated that the A-Star algorithm is suitable for use in cases where it is necessary to find a path quickly and well in various environments [4]. Septiana *et al.* in his research explained that the optimal path search process in the WSN routing technique using the A-Star algorithm is very dependent on the heuristic function, so that the effectiveness of the evaluation function is increased by adding a heuristic function when searching [5]. Whereas, Wang *et al.* in his research explained that a good path search algorithm is needed in navigating the path of travel, the A-Star algorithm is used to overcome low computation in map navigation and add a hierarchical mechanism to the path search problem [6].

Sabri *et al.* in his research revealed that Dijkstra's algorithm can find the shortest path in the case of choosing an exit route for the evacuation process. This algorithm was chosen because it can efficiently deduce the shortest path in its solution [7]. Abderrahim *et al.* in his research proposes a new energy saving algorithm using Dijkstra's algorithm, where this strategy can improve the results obtained [8]. Meanwhile, Jiang *et al.* In his research extending the famous shortest path algorithm to consider edge weights and node weights for graphs derived from SDN topology, the Dijkstra algorithm is applied in extended pyretics [9].

Several studies have shown that the Greedy, A-Star, and Dijkstra algorithms can be applied in the process of finding the shortest or fastest path in a path or graph. Each algorithm has its own advantages and disadvantages. This is what underlies the author to further analyze the Greedy, A-Star, and Dijkstra algorithms. The purpose of this research is to analyze the Greedy, A-Star, and Dijkstra algorithms in the process of finding the shortest path. The author wants to compare the effectiveness of the three algorithms in the process of finding the shortest path in a graph.

2. RESEARCH METHOD

2.1. Greedy Algorithm

Greedy Algorithm is an algorithm that can solve problems by making the best choices in certain cases. Many optimization problems can be solved using this algorithm [10]. The Greedy algorithm can reduce the computational complexity with the persistence of high mesh quality, however, this algorithm generates additional time costs, resulting from the selection loop [11]. Greedy algorithm can be combined in finding sub-optimal solutions in a shorter time [12]. The Greedy Algorithm is a gradual inference process starting from the zero model and solving the problem heuristically [13]. Greedy algorithm is recommended in signal processing and machine learning [13][14][15]. The Greedy Algorithm solves the problem step by step, at every step:

a. Take the best option available at that time.
b. Hope that by selecting local-optimum at each step will achieve global-optimum. This algorithm assumes that the local optimum is part of the global optimum.

The optimization problem in the context of the Greedy algorithm is composed of the following elements:

a. Candidate set (*c*).
b. The set of solutions (*s*).
c. Selection function.
d. Feasibility function.
e. Objective function.

In other words, the Greedy algorithm involves the search for a subset *s*, from the set of candidates *c*, in which case *s* must meet some specified criteria, namely declaring a solution and *s* being optimized by an objective function. Figure 1 shows an illustration of the Greedy algorithm

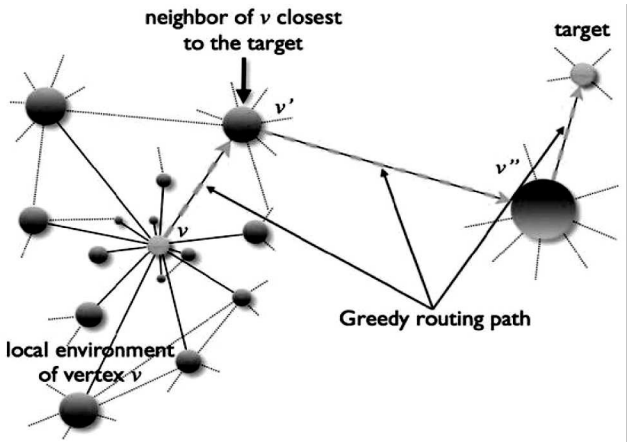


Figure 1. Greedy algorithm illustration

2.2. A-Star Algorithm

The A-Star algorithm is one of the best-known path planning algorithms [16], which can be applied to a configuration space metric or topology. This algorithm uses heuristic search and search based on the shortest path [17]. In terms of search, this algorithm is good in various environments [4]. The A-Star algorithm is a classic path search algorithm and can be used to search mazes [18]. The A-Star algorithm guides the optimal path to the goal if the heuristic function *h(n)* is acceptable, meaning that it will never overestimates the original cost [19] or actual cost [20].
Evaluation function $f(n) = g(n) + h(n)$, where [17][21][22]:
$g(n)$ = cost so far to reach n.
$h(n)$ = estimated cost from n to target (goal).
$f(n)$ = estimated total cost of the path through n to the target.

2.3. Dijkstra's Algorithm

Dijkstra's algorithm can be used to find the shortest path between nodes in a graph, for example on a road network or path. Djikstra's algorithm tends to be fast in finding solutions but is unable to deal with negative weight values [23]. Dijkstra's algorithm has a target, namely to find the distance from one node to another. This algorithm claims to be the best approach in solving the simple shortest path problem [7][24]. The Dijkstra algorithm for finding the shortest path is as follows[25]:

a. $L = \{\ \}$;
   $V = \{v_2, v_3, \ldots, v_n\}$
b. For $i = 2, \ldots, n,$ do $D(i) – w(1,i)$
c. As long as $v_n \notin L$ do:
   1) Select a point $v_k \in V – L$ with $D(k)$ is the smallest
      $L = L \cup \{v_k\}$
   2) For each $v_j \in V – L$ do,
      If $D(j) > D(k) + W(k,j)$ then change $D(j)$ with
      $D(k) + W(k,j)$
d. For each $v_j \in V$, $W^*(1,j) = D(j)$

2.4. Methodology

The purpose of this study is to analyze the Greedy, A-Star, and Dijkstra algorithms in the process of finding the shortest path and to compare the effectiveness of the three algorithms in the process of finding the shortest path in a path or graph. To achieve this goal, there are several processes or stages in analyzing the Greedy, A-Star, and Dijkstra algorithms in finding the shortest path. Figure 2 shows a diagram of the research method in the analysis of the Greedy, A-Star, and Dijkstra algorithms.

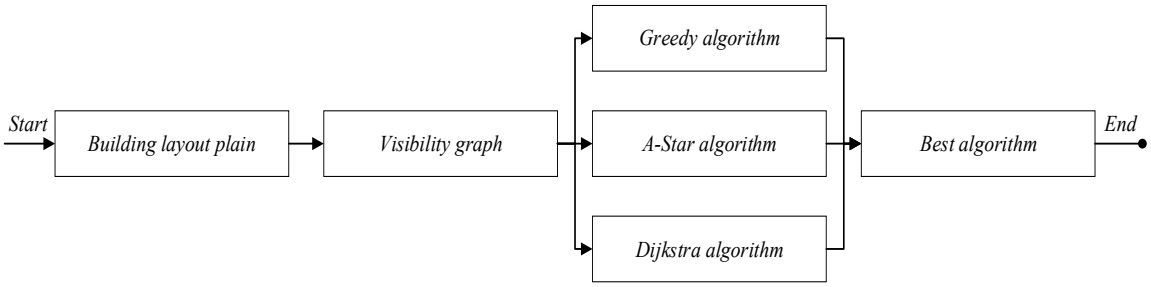


Figure 2. Research methodology diagram

In Figure 2, can be see the process of the Greedy, A-Star, and Dijkstra algorithm analysis flowchart in the process of finding the shortest path. Where the goal is to find out which algorithm is best in the process of finding the shortest path.

3. RESULTS AND DISCUSSION

The analysis process in comparing the Greedy, A-Star, and Dijkstra algorithms in the process of finding the shortest path requires a path or graph that already contains the cost or distance from one node to another. Figure 3 will illustrate the form of a path or graph that will be tested using the three algorithms discussed.

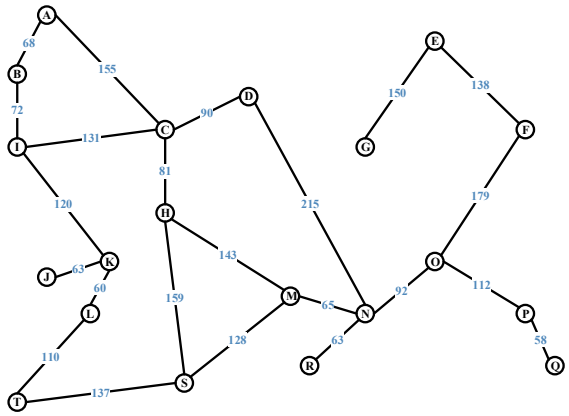


Figure 3. Tested path/graph

The next process is to set node A as the initial state and node N as the final state. Then apply the algorithm to the graph starting with testing the Greedy algorithm. In applying the Greedy algorithm, the results can be seen in Figure 4.

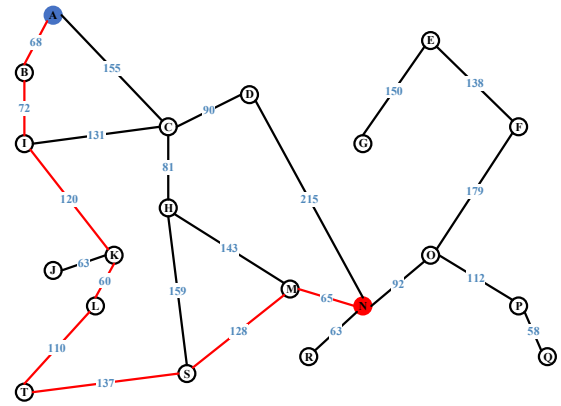


Figure 4. Testing the Greedy algorithm on graph

In Figure 4, it can be seen that the path taken from the initial state to the final state, the Greedy algorithm through a path that has a cost value that tends to be smaller so that the path that is

passed from the initial state to the final state is A → B → I → K → L → T → S → M → N, for a total cost of 760.

Next is to apply the A-Star algorithm to the graph with the same initial and final state as in the Greedy algorithm test, namely the initial state at node A and the final state at node N. However, first determine the straight-line distance to node N. The A-Star algorithm shows the results that can be seen in Figure 5.

Straight-line distance to node N

A	255	H	124	O	80
B	239	I	201	P	118
C	133	J	138	Q	121
D	131	K	139	R	53
E	198	L	130	S	126
F	164	M	58	T	189
G	98	N	0		

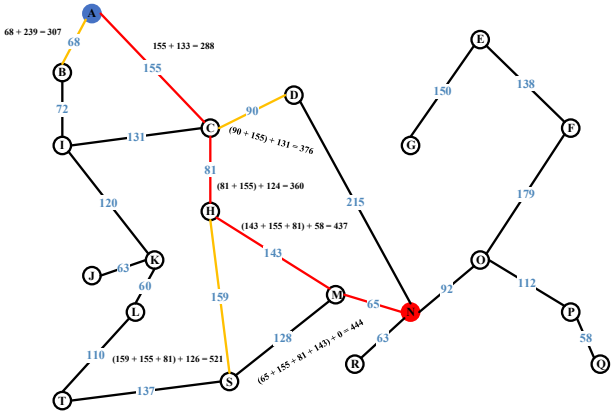


Figure 5. Testing A-Star algorithm on graph

In Figure 5, you can see the path that will be followed from the initial state to the final state. The A-Star algorithm goes through a path by calculating the cost of a path with straight-line distance to node N so that the path that is passed from the initial state to the final state is A → C → H → M → N, with a total cost of 444.

Next is to apply the Djikstra algorithm to the graph with the same initial and final state as in the Greedy and A-Star algorithm tests, namely the initial state at node A and the final state at node N. In applying the Dijkstra algorithm, the results can be seen in Figure 6.

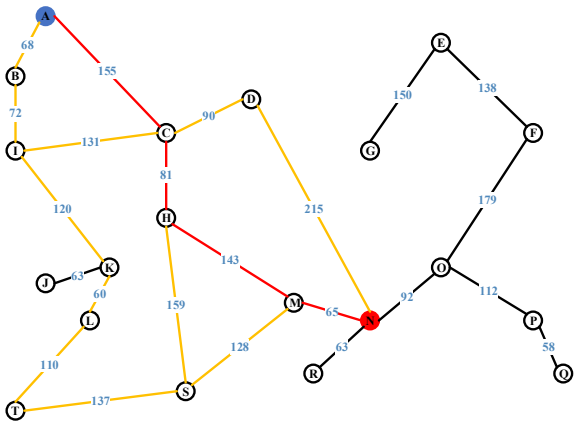


Figure 6. Testing Dijkstra's algorithm on graph

In Figure 6, you can see the path that will be followed from the initial state to the final state. Dijkstra's algorithm goes through a path by calculating the cost of a path by comparing each cost in the path from the initial state to the final. So that the shortest path that will be taken from the initial state to the final state is A → C → H → M → N, with a total cost of 444. Next, we retry the three algorithms by placing the initial state on a different node. Table 1 shows the trial results of the comparison of the Greedy, A-Star, and Dijkstra algorithms.

Table 1. Testing Algorithm Comparison on Graph

#	Initial → Final state	Cost		
		Greedy	A-Star	Dijkstra
1	A → N	760	444	444
2	B → N	512	692	492
3	C → N	289	289	289
4	H → N	386	386	208
5	K → N	500	500	500
6	P → N	*Not found*	*Not found*	204

In Table 1, we can see the comparison between the Greedy, A-Star, and Dijkstra algorithms. Where the difference in results for each algorithm is seen. In the Greedy and A-Star algorithms, there is a node that cannot be reached. Meanwhile, the Dijkstra algorithm can find all the target nodes but uses a very complex search. Figure 7 shows a comparison graph of the three algorithms.

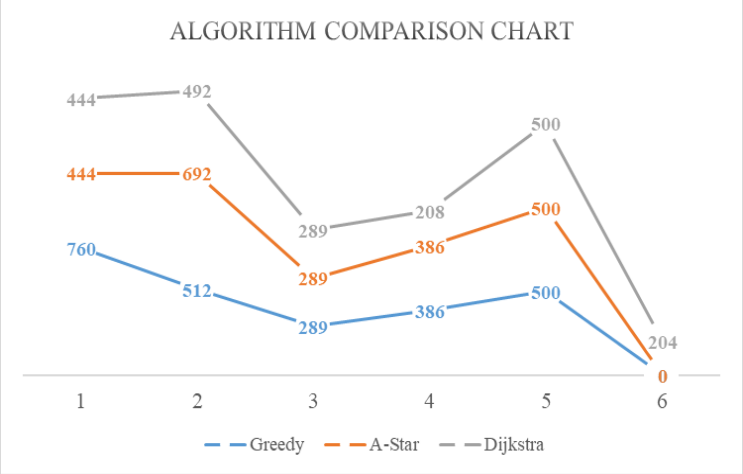


Figure 7. Algorithm comparison chart

In Figure 7, we can see a comparison between the Greedy, A-Star, and Dijkstra algorithms where the Dijkstra algorithm can find all the nodes that are targeted or destinations. So that from the three algorithms that were tested in the case of finding the shortest path with complex searches, the author recommends the Dijstra algorithm as a problem solution.

4. CONCLUSION

From the results of the research that has been done, several conclusions can be drawn, including:

a. The results of the Greedy algorithm analysis with several trials can find the shortest path in a fast time, but there are some cases where the optimal solution is not found or the solution (final state) is not found at all.
b. The results of the A-Star algorithm analysis with several trials can find the shortest path better than the Greedy algorithm, the second equation is that they have found cases where the target (final state) is not found.

c. The results of Dijkstra's algorithm analysis with several trials can find the shortest path better than the Greedy and A-Star algorithms. In this case Dijkstra's algorithm can find a solution (final state) which tends to be better than the two and always finds the optimal solution.
d. The weakness of the Greedy algorithm is that it tends to take choices that do not take into account the next event, while the weakness of the A-Star algorithm is that the graph must have complex data such as straight-line distance to node (final state), while the weakness of Dijkstra's algorithm is the opposite of the Greedy algorithm. that is, they tend to be slow in finding solutions because they have to compare the cost of one path with the cost of another path.

REFERENCES

[1] S. Bhosale & D. Kulkarni, "A Greedy Algorithm Approach for Mobile Social Network", *International Journal of Computer Applications, volume 111, number 16*, 2015.
[2] H.E. Zhenhuan, "Research on Improved Greedy Algorithm for Train Rescheduling", *IEEE 7th International Conference on Computational Intelligence and Security*, pp. 1197- 1200, 2011.
[3] B.M. Patil & B. Amarapur, "Segmentation of Leaf Images using Greedy Algorithm", *IEEE International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, pp. 2137-2141, 2017.
[4] F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico, & L. Jurisica, "Path Planning with Modified A Star Algorithm for a Mobile Robot", *Elsevier Modelling of Mechanical and Mechatronic Systems*, pp. 59-69, 2014.
[5] R. Septiana, I. Soesanti, N.A. Setiawan, "Evaluation Function Effectiveness in Wireless Sensor Network Routing using A-Star Algorithm", *IEEE 4th International Conference on Cyber and IT Service Management*, pp. 1-5, 2016.
[6] H. Wang, J. Zhou, G. Zheng, & Y. Liang, "HAS: Hierarchical A-Star Algorithm for Big Map Navigation in Special Areas", *IEEE International Conference on Digital Home*, pp. 222-225, 2014.
[7] N.A.M. Sabri, A.S.H. Basari, B. Husin, & K.A.F.A. Samah, "The Utilisation of Dijkstra's Algorithm to Assist Evacuation Route in Higher and Close Building", *Journal of Computer Science*, pp. 330-336, 2015.
[8] M. Abderrahim, H. Hakim, H. Boujemaa, & F. Touati, "Energy-Efficient Transmission Technique Based on Dijkstra Algorithm for Decreasing Energy Consumption in WSNs", *IEEE 19th International Conference on Sciences and Techniques of Automatic Control & Computer Engineering*, pp. 599-604, 2019.
[9] J.R. Jiang, H.W. Huang, J.H. Liao, & S.Y. Chen, "Extending Dijkstra's Shortest Path Algorithm for Software Defined Networking", *IEICE Asia Pasific Network Operation and Management Symposium (APNOMS)*, pp. 1-4, 2014.
[10] A. Malik, A. Sharma, & V. Saroha, "Greedy Algorithm", *International Journal of Scientific and Research Publications, volume 3, issue 8, ISSN 2250-3153*, pp. 1-5, 2013.
[11] R. Zhao, C. Li, X. Guo, S. Fan, Y. Wang, Y. Liu, & C. Yang, "A Parallel Iteration Algorithm for Greedy Selection Based IDW Mesh Deformation in OpenFOAM", *IEEE 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, pp. 1449-1452, 2019.
[12] V.P. Patel & R.K. Hardik, "Optimization of Large Join Query using Heuristic Greedy Algorithm", *International Journal of Computing and Technology (IJCAT), Volume 1, Issue 1*, 2014.
[13] L. Xu, S. Lin, J. Zeng, X. Liu, Y. Fang, & Z. Xu, "Greedy Criterion in Orthogonal Greedy Learning", *IEEE Transactions on Cybernetics*, pp. 1-12, 2017.
[14] H. Chen, Y. Zhou, Y.T. Tang, L. Li, & Z. Pan, "Convergence Rate of the Semi-supervised Greedy Algorithm", *Elsevier Neural Networks, vol. 44*, pp. 44-50, 2013.
[15] S.B. Lin, Y.H. Rong, X.P. Sun, & Z.B. Xu, "Learning Capability of Relaxed Greedy Algorithms," *IEEE Transactions on Neural Networks and Learning Systems*, *vol. 24*, pp. 1598–1608, 2013.
[16] K.B. Sung & D.Y. Kwak, "System Architecture for Autonomous Driving with Infrasturcture Sensors", *IEEE 6th International Conference on Signal Processing and Communication Systems*, pp. 1-6, 2012.
[17] P.O.N. Saian, Suyoto, & Pranowo, "Optimized A-Star Algorithm in Hexagon-Based Environment using Parallel Bidirectional Search", *IEEE 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 1-5, 2016.
[18] X. Liu & D. Gong, "A Comparative Study of A-Star Algorithms for Search and Rescue in Perfect Maze", *IEEE International Conference on Electric Information and Control Engineering*, pp. 1-4, 2011.

[19] S. Mahadewi, K.R. Shylaja, & M.E. Ravinandan, "Memory Based A-Star Algorithm for Path Planning of a Mobile Robot", *International Journal of Science and Research (IJSR), Volume 3, Issue 6*, pp. 1351-1355, 2014.
[20] A. Candra, M.A. Budiman, & K. Hartanto, "Dijkstra's and A-Star in Finding the Shortest Path: a Tutorial", *IEEE International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, pp. 28-32, 2020.
[21] J. Yao, C. Lin, X. Xie, A.J. Wang, & C.C. Hung, "Path Planning for Virtual Human Motion using Improved A* Algorithm", *IEEE 7th International Conference on Information Technology*, pp. 1154-1158, 2010.
[22] S. Sedighi, D.V. Nguyen, & K.D. Kuhnert, "Guided Hybrid A-Star Path Planning Algorithm for Valet Parking Applications", *IEEE 5th International Conference on Control, Automation, and Robotics*, pp. 570-575, 2019.
[23] T. Surekha & R. Santosh, "Review of Shortest Path Algorithm", *International Research Journal of Engineering and Technology (IRJET), Volume 3, Issue 8*, pp. 1956-1959, 2016.
[24] Risald, A.E. Mirino, & Suyoto, "Best Routes Selection using Dijkstra and Floyd-Warshall Algorithm", *IEEE International Conference on Information & Communication Technology and System (ICTS)*, pp. 155-158, 2017.
[25] Y. Chao, "A Developed Dijkstra Algorithm and Simulation of Urban Path Search", *IEEE 5th International Conference on Computer Science & Education*, pp. 1164-1167, 2010.