ENHANCED KNOWLEDGE IN SCIENCES AND TECHNOLOGY
e-ISSN: 2773-6385

EKST

Vol. 6 No. 1 (2026) 125-131
https://publisher.uthm.edu.my/periodicals/index.php/ekst

Analysis of Dijkstra's and A* Algorithms for the Shortest Route Problem

Muhammad Syafiq Azri Muhamad Azhar [1], Muhamad Ghazali Kamardan[1]*

[1] *Department of Mathematics and Statistics, Faculty of Applied Sciences and Technology, UTHM Kampus Cawangan Pagoh, Hab Pendidikan Tinggi Pagoh, KM 1, Jalan Panchor, 84600 Pagoh, Muar, Johor, MALAYSIA.*

*Corresponding Author: mghazali@uthm.edu.my
DOI: https://doi.org/10.30880/ekst.2026.06.01.016

Article Info

Received: 2 January 2026
Accepted: 5 June 2026
Available online: 9 July 2026

Keywords

Shortest Route Problem, Djikstra's Algorithm, A* Algorithm, KSL City Mall

Abstract

Finding an efficient route between two locations is a common need for users in modern navigation systems especially in busy and crowded city such as Johor Bahru. Navigating efficiently from one location to another is often challenging because of complex road networks and multiple intersections. To solve this, many navigation systems use algorithms such as Dijkstra's algorithm and A* algorithm to determine the best and quickest route. However, different algorithms work differently, and some are faster or more effective than others. This study presents a comparison between Dijkstra's algorithm and the A* algorithm in solving the Shortest Route Problem (SRP) between JB Sentral and KSL City Mall in Johor Bahru, Malaysia. Road data network was collected from Google Maps and converted into a weighted graph, where road intersections are nodes and road segments represented as edges with distance values in. Both algorithms were implemented using Python and evaluated based on the total route distance and the number of iterations required to reach the destination. The A* algorithm completes the search using fewer iterations which is 7 iterations compared to Dijkstra's algorithm that needs 11 iterations to reach the destination, showing the A* algorithm has better efficiency. Even though Djikstra's algorithm is still accurate and reliable, A* algorithm is more preferred for applications that need a faster route computation.

1. Introduction

Navigation applications play an important role in helping users identify the shortest and most efficient routes between locations. In Johor Bahru, Malaysia, one of the most frequently travelled routes is between JB Sentral, the city's main transportation hub, and KSL City Mall, a major commercial destination. Optimizing this route is beneficial for tourists, commuters and urban transportation planning. One of the major issues is that complex road network and numerous crossroads leave users struggling to select optimal shortest path. As defined in [1], the dominant problem these systems focus on solving is the (SRP), which seeks a path between two nodes through a computer network. Furthermore, the (SRP) can be formulated as the shortest path problem between two nodes or a quickest path from one node to another [2]. This can be formulated with graph theory, by creating a node for the locations and edges as roads with weights equal to distances. The purpose of this research is to contrast the efficiency and accuracy of Djikstra algorithm and A* algorithm in seeking shortest path. This serves as a mechanism to establish the algorithm which provides better results with real applications.

The (SRP) is the process of finding a lowest cost path between two nodes in a network and is commonly used within area such as Navigation systems, traffic networks and route optimization explored by [1]. [3] proved that the most widely used algorithms for solving (SRP) are Dijkstra's algorithm and A* algorithm. Dijkstra's algorithm

was first published by Edsger W. Dijkstra in 1959, is a greedy algorithm which ensures the shortest path for non-negative edge-weighted graphs through exploring nodes with minimum known distance from the source at any point of time. Djikstra's algorithm is commonly used in practical real-world example such as GPS navigation and even scientific research area as mentioned in [4]. In contrast, A* algorithm is an extension of Dijkstra's to support heuristics, which allows it to achieve faster performance if the search can be steered toward the goal [5]. As reported in [6], the A* algorithm is an upgraded version of Dijkstra's for more efficient path finding and implements heuristic knowledge within the search for the path. A* Algorithm starts with the best-first algorithm that selects the lowest-cost path by using two cost values which is a cost from the start to the current node and an estimated cost to goal. This algorithm is popular in real-time pathfinding and navigation games because it is very efficient, as well as flexible for a very wide range of possible applications. This paper compares the performance of Dijkstra's algorithm and A* algorithm in finding the shortest route from JB Sentral to KSL City Mall which is using an actual road data that extracted from Google Maps.

2. Methodology

The SRP problem has attracted significant attention due to its application to transport, logistics and navigation systems. It is the problem of finding the minimum total length path between two nodes in a graph [7]. Many algorithms have been proposed over the year to address this problem and some of the most widely used methods include Dijkstra's algorithm and the A* algorithm. The aim of this study is to resolve the (SRP) and compare Dijkstra's Algorithm with A* Algorithm in accuracy and efficiency. The description of the method follows in this section. It begins with data gathering in section 2.1, followed by Djikstra's Algorithm in section 2.2 and then section 2.3 which describes the A* Algorithm:

2.1 Data Collection

This research focuses on the road network between JB Sentral and KSL City Mall, located in Johor Bahru, Johor, Malaysia. JB Sentral functions as the main transportation hub in the city, while KSL City Mall is a major commercial center. The selected study area represents urban road environment with multiple intersection, which introduces multiple path choices to make it suitable for evaluating shortest path algorithms. This is a common route that frequently used by visitors and tourists, making it suitable case for shortest route analysis.

The selected study area represents an urban environment with multiple road intersections such as JB Sentral, Jalan Tun Abdul Razak, Jalan Lingkaran Dalam, Jalan Stesen, Jalan Tengku Azizah, Jalan Bukit Chagar, Danga City Mall, Jalan Kebun Teh, Jalan Dato Abdullah Tahir, Jalan Harimau and KSL City Mall. These locations were chosen because it is the most common route used by visitors and tourist. The road data network was collected from Google Map to provides reliable real world road information. From this graph model, each road intersection was represented as a node and the road segments were represented as edge. Each edge was assigned a weight based on the actual distance in kilometres.

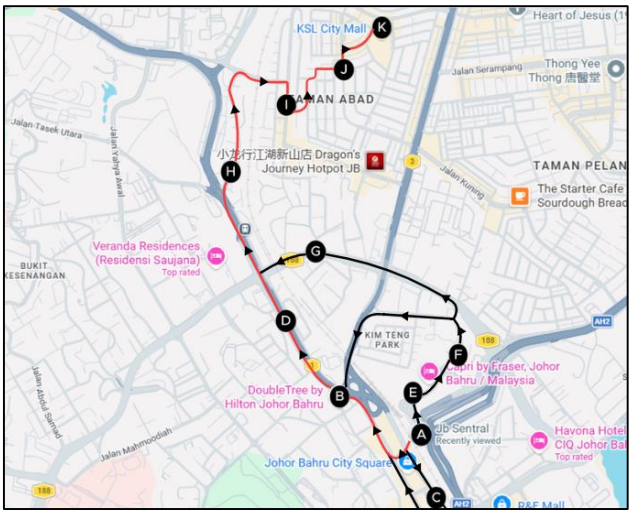

Fig. 1 Road Network between JB Sentral and KSL City Mall

Fig. 1 shows the route of selected study area road network between JB Sentral and KSL City Mall in Johor Bahru. The map shows the real-world road layout using Google Maps and highlights the possible routes connecting the starting point at JB Sentral to destination point at KSL City Mall. The labelled point on the map represents a key location or road intersection within the study area, which is modelled as node in the graph.

Enhanced Knowledge in Sciences and Technology Vol. 6 No. 1 (2026) p. 125-131

The connecting road segments between nodes represent edges with directional arrows indicating the allowable direction of travel. Distance of each road segment is measured in kilometres then assigned as edge weights to ensure accurate representation of real-world map. Red highlighted routes represent the shortest route from starting node to destination node. This figure provides a visual reference for the graph-based model used in comparison of Djikstra's algorithm and A* algorithm in this study.

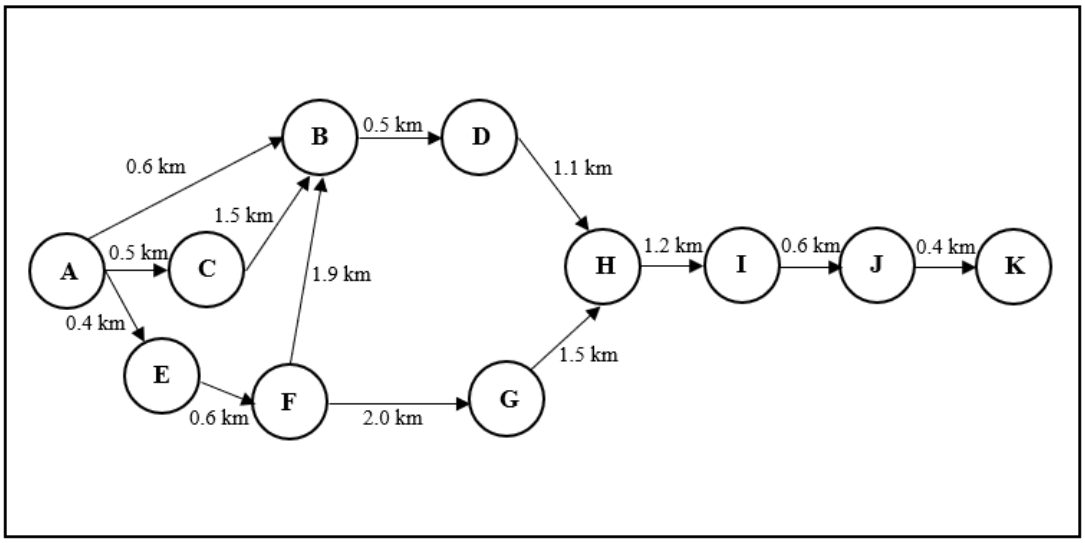

Fig. 2 Nodes in the Graph Model

Fig. 2 illustrates the graph model constructed from real-world road network data. This graph model transforms from physical road network into a mathematical structure that can be easier for users to understand. It contains 11 nodes and each of them are connected by road segments which can be labelled as edge. Every edge was measured by actual distance in kilometres. The model is used to compare the shortest route results from Djikstra's algorithm and A* algorithm from JB Sentral which is the starting point to destination point KSL City Mall.

Table 1 shows the list of nodes and their real-world locations, while Table 2 shows the distances between connected nodes in kilometres. Djikstra's algorithm and A* algorithm were used to determine the shortest possible route based on these distance values. This study indicates a fair and realistic comparison of both algorithms under the same conditions by modelling the road network in the structured way.

Table 1 Node and Edge from JB Sentral to KSL City Mall

Node	Location
A	JB Sentral
B	Jalan Tun Abdul Razak
C	Jalan Lingkaran Dalam
D	Jalan Stesen
E	Jalan Tengku Azizah
F	Jalan Bukit Chagar
G	Danga City Mall
H	Jalan Kebun Teh
I	Jalan Dato Abdullah Tahir
J	Jalan Harimau
K	KSL City Mall

Table 2 Distance of each node between JB Sentral to KSL City Mall

From	To	Distance (km)
A	B	0.6
A	C	0.5
A	E	0.4
B	D	0.5
C	B	1.5
D	H	1.1
E	F	0.6
F	B	1.9
F	G	2.0
G	H	1.5
H	I	1.2
I	J	0.6
J	K	0.4

2.2 Dijkstra's Algorithm

Dijkstra's algorithm operates by gradually expanding from the source node to other nodes in the graph investigated by [8]. At each step, the node with the smallest temporary distance value is selected and its neighbouring nodes are updated. This process continues until the destination node is reached. The graph is expressed as a $(n\ x\ n)$ matrix $D = [d_{ij}]$, where d_{ij} represents the weight or distance of the edge from vertex i to vertex j. If there is no edge between i and j, d_{ij} is considered infinite [9]. The algorithm is trying to calculate the smallest distance between a source node and a destination node. The algorithm works by keeping the shortest distance of each vertex from the source in a label region. The source vertex is given the distance value of zero and all other vertices are assigned a distance of infinity. Each round, the node with the shortest temporary distance is selected and its distance is set to permanent. Relaxation then updates the distances of its adjacent vertices. We update distance to vertex for each edge using:

$$[u_j, i] = [u_i + d_{ij}, i], when\ d_{ij} > 0 \quad (1)$$

Where u_i is the shortest known distance to vertex i, and d_{ij} is the weight of the edge connecting i and j. This process continues until the destination vertex receives its final distance or all vertices have been processed. Once completed, the algorithm guarantees the shortest path between the source and destination nodes. Due to its systematic exploration of all reachable nodes, Dijkstra's algorithm ensures optimal results but may explore more nodes compared to heuristic-based methods, especially in large graphs.

2.3 A* Algorithm

A* algorithm technique popular for road network applications, it is used to get responsive and fast results such as digital map and the book of navigation system available for tourist users [10]. An algorithm uses a heuristic function in order to estimate the remaining distance to the destination so that it can minimize an unnecessary route search [11]. This enables us to search for the minimum path with the optimal latency time very fast [12]. The algorithm computes the cost for each node based on a function that sums the distance to reach it from the start node and its estimated future distance to the goal. B: When selecting nodes observed, it considers those closer to the destination; this way unnecessary exploration is avoided and efficiency is improved exactly when dealing with larger networks. The A* algorithm ranks each node with a cost function, which is:

$$f(n) = g(n) + h(n) \quad (2)$$

where $g(n)$ represents the actual cost from the starting node to the current node, and $h(n)$ is a heuristic estimate of the remaining cost from the current node to the destination. Using true and predicted costs, the algorithm prefers nodes that are most likely form an optimal path.

The algorithm keeps two lists of nodes: open, which is a list of nodes to be evaluated and closed, which contains the already-evaluated nodes. And to begin with, the initial node is put into the open list. At each iteration, the node with the lowest $f(n)$value is selected, moved to the closed list, and its neighboring nodes are examined.

Enhanced Knowledge in Sciences and Technology Vol. 6 No. 1 (2026) p. 125-131

The costs for each neighbour are computed and if a path to a neighbour node was shorter than any previous practice shortened. This process is repeated until we have reached the destination node.

In this realization, the road in between JB Sentral and KSL City Mall are drawn from Google Maps distance data. Nodes representing critical junctions or choice points (intersections and decision nodes) were defined along the route, with connecting highways modelled as edges with cost measured in kilometres. The route was modelled using 11 nodes. The two algorithms were analysed and compared with respect to the total distance received and how many instances it takes to find a solution.

The test was static so no dynamics, like traffic jams or blocked roads, are in action. All performance comparisons were measured according to the total distance of the route and the number of loop counts, which indicate how many iterations or nodes are visited.

The road network was represented as a weighted undirected graph of 11 nodes, corresponding to major intersections and landmarks between the source and target locations, while edges were traffic segments weighted by distance in km. As all edge weights were non-negative, both Dijkstra's and A* algorithms could be used. The abstraction model offers simulation of the Johor Bahru urban road structure in a realistic and controlled manner.

3. Results and Discussion

This section presents the results obtained from the implementation of Dijkstra's algorithm and the A* algorithm for solving the Shortest Route Problem (SRP) between JB Sentral and KSL City Mall in Johor Bahru, Malaysia. Table 3 shows that both algorithms successfully identified the shortest route between JB Sentral and KSL City Mall with a total distance of 4.4 km. This observation was independent of the number of runs, using to certainty´s both approaches. Nevertheless, there were differences in search performance. Djikstra's algorithm needed 11 iterations to complete the search, while A* algorithm did it in just 7 iterations. Direct comparisons between the two algorithms indicate that while they generated identical route distances, it was observed that A* algorithm performed more efficiently as it decreased the number of iterations when compared to Dijkstra's algorithm. The smaller number of iterations indicates that A* might be more scalable in bigger and more complex road networks. The shortest path is about 4.4 km, which demonstrates the map abstraction of graph as well as the specific selection of main roads that are used, and is not inconsistent with reported driving distances from navigation applications where there are possibly detours included or alternative routes suggested.

Table 3 Comparative Summary

Metric	Dijkstra's Algorithm	A* Algorithm
Distance (km)	4.4	4.4
Loop Count	11	7

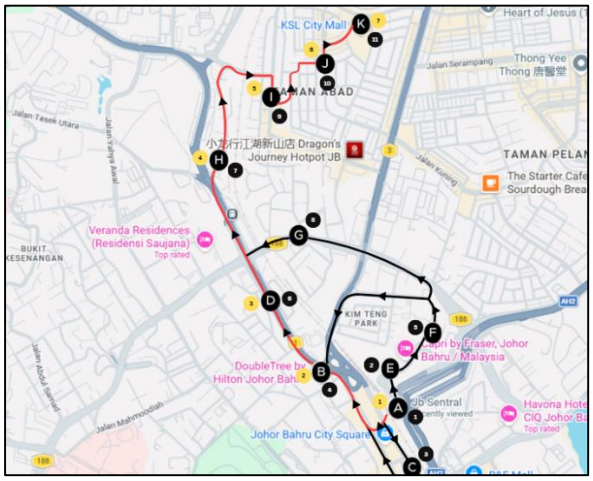

Fig. 3 Comparison of iterations between Dijkstra's algorithm and A* algorithm

Fig. 3 shows the iterations process and route exploration of the shortest route algorithm using two different algorithm which is A* algorithm and Dijkstra's algorithm. The figure shows how routes are travel from one iteration to another iteration starting from JB Sentral towards KSL City Mall. The black labelled numbers show the

iterations process of Dijkstra's algorithm which explores a large number of nodes. This because it systematically evaluates all possible paths on increasing distance value which effect a higher loop count. However, the yellow labelled numbers illustrate the process of A* algorithm to reach the destination point. A* algorithm successfully reach the optimal route with fewer iterations due to heuristic guidance that allow it to selected nodes that is closer to destination. Overall, both algorithms successfully produce the same shortest distance while A* algorithm performs better than Dijkstra's algorithm in terms of efficiency.

4. Conclusion

This study has compared the performance of Dijkstra's algorithm and A* algorithm to solve Shortest Route Problem at KSL City Mall and JB Sentral with real road network data. The route was represented as a weighted graph representing the google maps distance in km, and both algorithms were implemented in python. Results indicate that the two algorithms were able to find a shortest path with a distance of 4.4 km ensues in both methods. This illustrates that Dijkstra's algorithm and the A * algorithm is accurate and reliable for solving shortest path problems in road networks with non-negative edge costs. The use of a heuristic function in the A* algorithm did not affect the correctness of the solution.

However, a difference was observed in terms of efficiency. Dijkstra's algorithm visited more nodes than the A* algorithm. Dijkstra's algorithm needs 11 iterations and only 7 iterations for the A* algorithm to arrival at destination. The reason was that the A* algorithm explored less numbers of nodes because it was guided by heuristics, and making it faster than Dijkstra's algorithm to reach the destination. This demonstrates that the A* algorithm is more efficient in terms of search process, therefore this prove that A* algorithm successfully over performing Dijkstra's algorithm in terms of loop counts.

In conclusion, both algorithms are suitable for shortest route computation, the A* algorithm is more effective for navigation and route-planning applications, particularly in larger urban environments where reducing unnecessary node exploration is important.

5. Recommendation

Future studies may consider to expand the study area to include a larger and complex road network by adding more intersection and nodes. By expanding the network, it will help demonstrate how each algorithm performs as the size increase. This will highlight the better efficiency of A* algorithm in large urban environment. In addition, it is also possible to include comparison with another shortest path algorithm such as Bellman-Ford and Floyd-Warshall. This comparison would allow researchers to identify strengths and weakness of each algorithm in different scenarios.

Acknowledgement

The authors would like to express their sincere appreciation to the Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia, for its support.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design****: Muhammad Syafiq Azri Muhamad Azhar, Muhamad Ghazali;* ***data collection****: Muhammad Syafiq Azri Muhamad Azhar;* ***analysis and interpretation of results****: Muhammad Syafiq Azri Muhamad Azhar, Muhamad Ghazali****; draft manuscript preparation****: Muhammad Syafiq Azri Muhamad Azhar. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] X. Z. Wang, "The comparison of three algorithms in shortest path issue," *Journal of Physics Conference Series*, vol. 1087, p. 022011, Sep. 2018, doi: 10.1088/1742-6596/1087/2/022011.

[2] D. Rachmawati and L. Gustin, "Analysis of Dijkstra's algorithm and A* algorithm in shortest path problem," *Journal of Physics Conference Series*, vol. 1566, no. 1, p. 012061, Jun. 2020, doi: 10.1088/1742-6596/1566/1/012061.

[3] N. Gupta, K. Mangla, A. K. Jha, & M. Umar, "Applying Dijkstra's algorithm in routing process," *Umar International Journal of New Technology Research*, 2(5), 122–124. 2016 Available: https://l1nq.com/mZAms

[4] B. Cevizci, "Calculating the shortest path using Dijkstra's algorithm." https://eric.ed.gov/?id=EJ1341857

[5] V. Yevsieiev, A. Abu-Jassar, and S. Maksymova, "Building a traffic route taking into account obstacles based on the A-star algorithm using the python language" 2024. https://openarchive.nure.ua/handle/document/26244
[6] M. H. P. Yudha, S. Supian, and H. Napitupulu, "Optimalization route to tourism places in West Java using A-STAR algorithm," *CAUCHY Jurnal Matematika Murni Dan Aplikasi*, vol. 7, no. 3, pp. 464–473, Oct. 2022, doi: 10.18860/ca.v7i3.17032.
[7] K. Magzhan, & H. M. Jani, "A review and evaluations of shortest path algorithm," *International Journal of Scientific & Technology Research*, 2(6), 99–104, 2013 Available: https://surl.lt/bvaflp
[8] M. A. Javaid, "Understanding Dijkstra Algorithm," *SSRN Electronic Journal*, Jan. 2013, doi: 10.2139/ssrn.2340905.
[9] N. Ojekudo, Akpofure, Akpan, and N. Paul, "Anapplication of Dijkstra's Algorithm to shortest route problem," *IOSR Journal of Mathematics*, vol. 13, no. 1, pp. 20–32, 2017, doi: https://doi.org/10.9790/5728-1303012032.
[10] M. E. Miyombo *et al.*, "Optimal path planning in a real-world radioactive environment: A comparative study of A-star and Dijkstra algorithms," *Nuclear Engineering and Design*, vol. 420, p. 113039, Feb. 2024, doi: 10.1016/j.nucengdes.2024.113039.
[11] G. Chen, T. Wu, and Z. Zhou, "Research on ship meteorological route based on A-Star algorithm," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–8, May 2021, doi: 10.1155/2021/9989731.
[12] R. Ropiqoh and R. S. Lubis, "Implementation of A* algorithm in finding the shortest route of cooking oil distribution in Karo Regency using graph," *Mathline: Jurnal Matematika dan Pendidikan Matematika*, vol. 8, no. 2, pp. 725–738, 2023. https://doi.org/10.31943/mathline.v8i2.442