IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

COMPARITIVE ANALYSIS ON SHORTEST PATH COMPUTATION IN GPS SYSTEMS

Sen Oommen James[1], Dr. Anjana S Chandran[2]

[1]Scholar, SCMS, Cochin, Kerala, India,

[2] Assistant Professor, SCMS, Cochin, Kerala, India,

Abstract: Distributed computing is a field of computer science that deals with the computation of distributed systems. The use of concurrent processes which was communicated by the process of message-passing has their own developments in operating system architectures. This paper discusses the results of monitoring, comparing and analyzing various algorithms which are used to find the shortest path between two locations in a GPS technology. The results are based on comparing various factors such as CPU utilization, CPU load, RAM utilization, network load, accuracy of the model and cross entropy value. The results are then discussed for various insights that are obtained over the course of this project.

Keywords:GPS,GlobalPositioningSystem

CPU,Centralprocessingunit

RAM,Randomaccessmemory

1. Introduction and Background Study

Distributed systems are by now very common, yet remains a difficult area of research. The results of new technologies coming up is that it's not only feasible, but easy, to hold together a computer system packed of multiple network computers, be they too large or too small. These computers are generally geographically dispersed, that reason they're usually said to make a distributed system. The size of a distributed system may vary from a couple of devices, to many computers. The interconnection network could also be wired, wireless, or a mixture of both. Moreover, distributed systems are often highly dynamic, within the sense that computers can join and leave, with the topology and performance of the underlying network almost continuously changing.

The Global Positioning System (GPS), firstly named as NAVSTAR GPS, is a satellite-based radio-navigation system which is owned by the US government and operated by the USSF (United States Space Force).GPS is a location based system which has 30 plus satellites surrounding the earth. GPS has mainly 3 components satellites, ground stations and a receiver. The satellites emit the signals continuously and the receiver in a device is constantly monitoring the result. The ground stations where made to make sure the correctness in satellites location by the use of radar. If a request is given from a device say X, the X calculates its distance to four or more satellites by figuring out how long it took for the signals to reach it. If the receiver knows its distance from four satellites it's easy to identify X's geolocation in co-ordinates. GPS apps are not only used by civilians but also by different organizations for different purposes. Different applications like Aviation, Marine, Farming, Science, Financial Services , Surveying, Military, , Telecommunications, Heavy Vehicle Guidance, Road Transportation, Social Activities, Locating Positions, GPS technology combined with Intelligence Vehicle Highway Systems is used to improve highway safety and ease congestion, Public Safety and Disaster Relief, Setting up a Geo-Fence etc.. . Out of which searching a location and finding the appropriate path is the commonly used application.

For finding the shortest ad appropriate path we have many algorithms like Dijkstra Algorithm, Breadth first algorithm, Depth first algorithm and A* heuristic algorithm.

2. PROBLEM STATEMENT AND OBJECTIVES

The problem definition of the context is to monitor, compare and analyse various algorithms based on shortest path finding in a GPS technology with different architectures under similar environment to provide insights regarding the performance and resource requirements by them.

The major objectives are as follows:

1. The objective of the paper was to determine which search method is suitable for implementation in GPS systems.
2. The properties of path finding algorithms were tested and discussed taking into account this type of systems. 3. Compare and analyse various results obtained as part of the testing and the results obtained by monitoring the system environment.

3. CONCEPTS & METHODOLOGY

The shortest path problem is about finding a path between two vertices in a graph such that the total sum of the edges weights is minimum. The problem of finding the shortest path between two intersections on a road map may be modelled as a special case of the shortest path problem in graphs, where the vertices correspond to intersections and the edges correspond to road segments, each weighted by the length of the segment. Shortest path algorithms are applied to automatically find directions between physical locations, such as driving directions on web mapping websites like MapQuest or Google Maps.

For this application fast specialized algorithms are available. If one represents a nondeterministic abstract machine as a graph where vertices describe states and edges describe possible transitions, shortest path algorithms can be used to find an optimal sequence of choices to reach a certain goal state, or to establish lower bounds on the time needed to reach a given state. For example, if vertices represent the states of a puzzle like a Rubik's Cube and each directed edge corresponds to a single move or turn, shortest path algorithms can be used to find a solution that uses the minimum possible number of moves. In a networking or telecommunications mind-set, this shortest path problem is sometimes called the min-delay path problem and usually tied with a widest path problem. For example, the algorithm may seek the shortest (min-delay) widest path, or widest shortest (min-delay) path.

A more light-hearted application is the games of "six degrees of separation" that try to find the shortest path in graphs like movie stars appearing in the same film. Other applications, often studied in operations research, include plant and facility layout, robotics, transportation, and VLSI design

3.1. SHORTEST PATH ALGORITHMS USED

There are n numbers of algorithms available for computing the shortest path problem. In this study we compare mainly 4 types of algorithms.

3.1.1 A* ALGORITHM

A* (pronounced "A-star") is a graph traversal and path search algorithm, which is often used in computer science due to its completeness, optimality, and optimal efficiency. One major practical drawback is its space complexity, as it stores all generated nodes in memory. Thus, in practical travel-routing systems, it is generally outperformed by algorithms which can pre-process the graph to attain better performance, as well as memory-bounded approaches; however, A* is still the best solution in many cases. A* was originally designed for finding least-cost paths when the cost of a path is the sum of its edge costs, but it has been shown that A* can be used to find optimal paths for any problem satisfying the conditions of a cost algebra. A* is an informed 21 search algorithm, or a best-first search, meaning that it is formulated in terms of weighted graphs: starting from a specific starting node of a graph, it aims to find a path to the given goal node having the smallest cost (least distance travelled, shortest time, etc.). It does this by maintaining a tree of paths originating at the start node and extending those paths one edge at a time until its termination criterion is satisfied

3.1.2 DIJKSTRA ALGORITHM

Dijkstra algorithm (or Dijkstra Shortest Path First algorithm, SPF algorithm) is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by computer scientist Edger W. Dijkstra in 1956 and published three years later. The algorithm exists in many variants. Dijkstra original algorithm found the shortest path between two given nodes, but a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest-path tree. For a given source node in the graph, the algorithm finds the shortest path between that node and every other. It can also be used for finding the shortest paths from a single node to a single destination node by stopping the algorithm once the shortest path to the destination node has been determined. For example, if the nodes of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road (for simplicity, ignore red lights, stop signs, toll roads and other obstructions), Dijkstra algorithm can be used to find the shortest route between one city and all other cities. A widely used application of shortest path algorithm is network routing protocols, most notably IS-IS (Intermediate System to Intermediate System) and Open Shortest Path First (OSPF). It is also employed as a subroutine in other algorithms such as Johnson's.

3.1.3 BREADTH FIRST SEARCH ALGORITHM

Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key'), and explores all of the neighbour nodes at the present depth prior to moving on to the nodes at the next depth level. It uses the opposite strategy as depth-first search, which instead explores the node branch as far as possible before being forced to backtrack and expand other nodes. BFS and its application in finding connected components of graphs were invented in 1945 by Conrad Zeus, in his (rejected) Ph.D. thesis on the Plankalkül programming language, but this was not published until 1972. It was reinvented in 1959 by Edward F. Moore, who used it to find the shortest path out of a maze, and later developed, by C. Y. Lee into a wire routing algorithm (published 1961).

3.1.4 DEPTH FIRST SEARCH ALGORITHM

Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking. A version of depth-first search was investigated in the 19th century by French mathematician Charles Pierre Trémaux as a strategy for solving mazes. The time and space analysis of DFS differs according to its application area. In theoretical computer science, DFS is typically used to traverse an entire graph. For applications of DFS in relation to specific domains, such as searching for solutions in artificial intelligence or web-crawling, the graph to be traversed is often either too large to visit in its entirety or infinite (DFS may suffer

from nonterminating). In such cases, search is only performed to a limited depth; due to limited resources, such as memory or disk space, one typically does not use data structures to keep track of the set of all previously visited vertices.

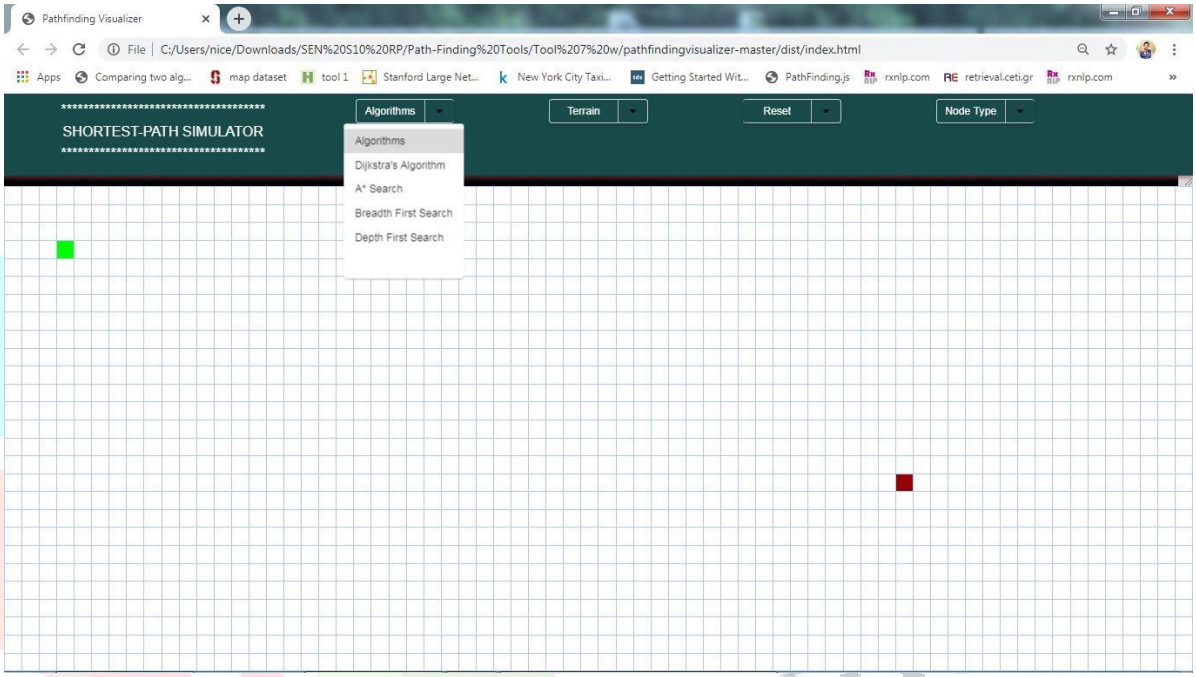
4. TOOLS AND PLATFORM USED

Platform used for the project is a shortest path computing visualizer with JavaScript libraries.
Various architectures used during the comparative study are listed below.
1. Shortest-path Simulator
2. JavaScript
3. React js.

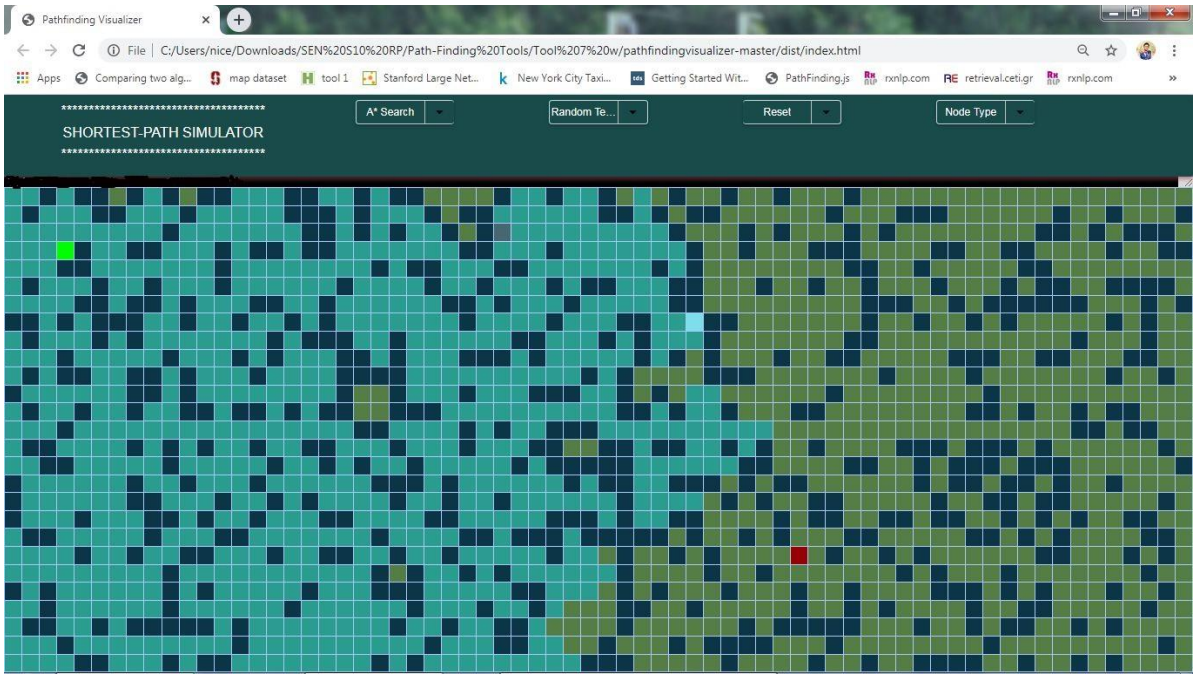
5. EXPERIMENT IMPLEMENTATION

The experimentation system was created in order to properly investigate the properties of the path finding algorithms. It contains the programmed simulator with the four implemented algorithms and three different terrain patterns.

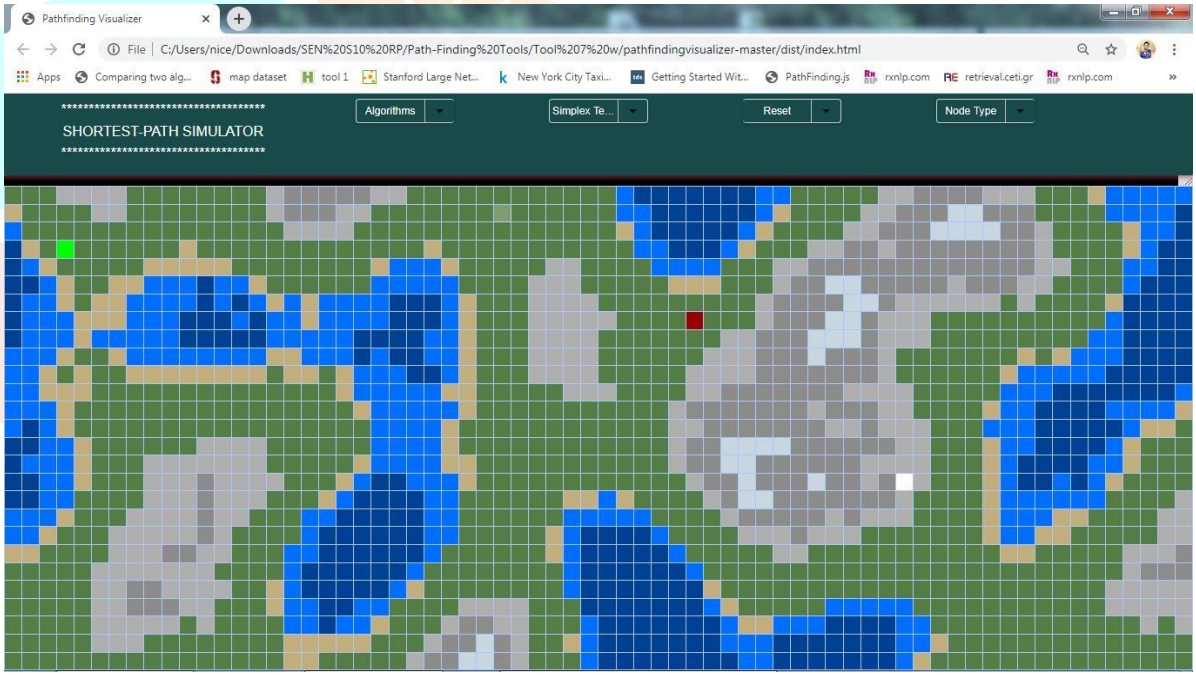
5.1. Normal Terrain



5.2. Random Terrain



5.3. Recursive Terrain



6. RESULTS

The results obtained during the simulation process of finding the shortest path between the source node and destination node are provided in this section. The results are in terms of final test accuracy, total number of Nodes visited, total Time taken for completing the process, total path Cost, CPU utilization, CPU load, memory utilization and network traffic. All of the above mentioned parameters has been measured for all four algorithms which are A*, Dijkstra, Breadth First and Depth First algorithms.

6.1 TEST ACCURACY USING SIMULATOR

The final test accuracy value is the taken according to the results obtained during the simulation process under multiple circumstances. The final value taken when multiple parameters like number of nodes visited, time taken and path cost. This value are significant because it represents the accuracy or in other words rate of successful shortest-path finding process. The following figures represent the final test results of the various algorithms. Each algorithm will have 3 different test accuracy results according to the different terrains.

6.1.1NORMALTERRAIN

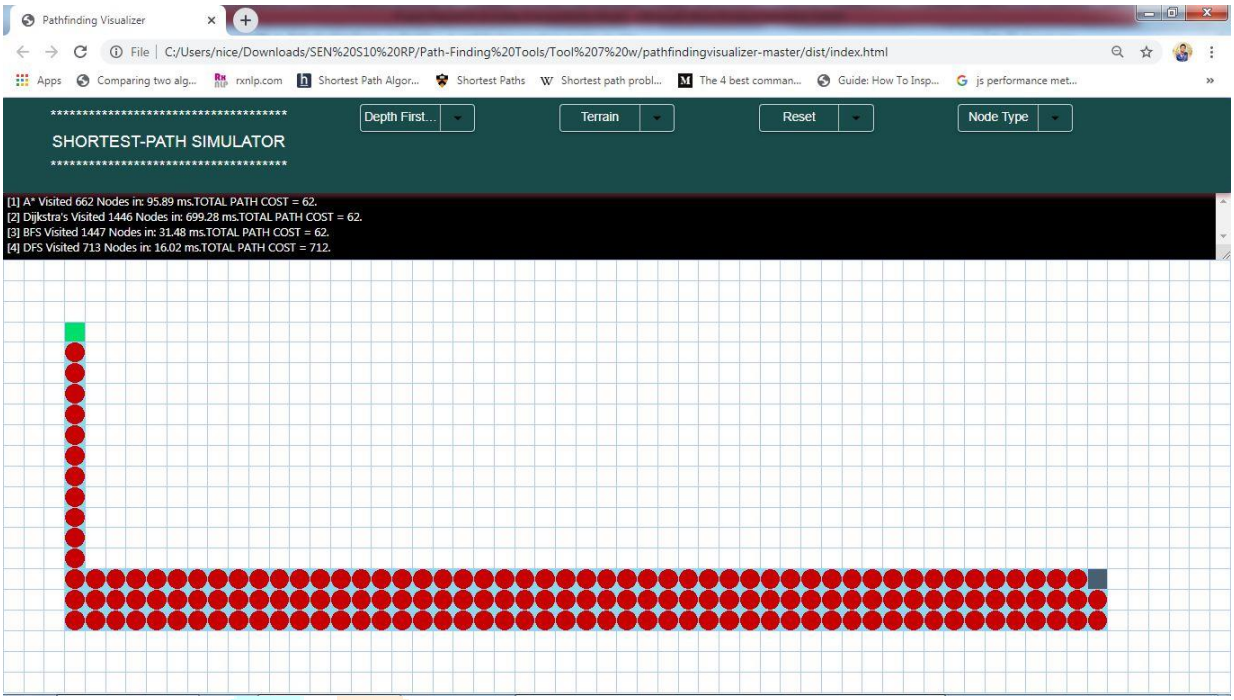


Figure 6.*1*.1: RESULT *SET OF ALL ALGORITHMS UNDER NORAML TERRAIN CONDITION.*

6.1.2 RECURSIVE TERRAIN

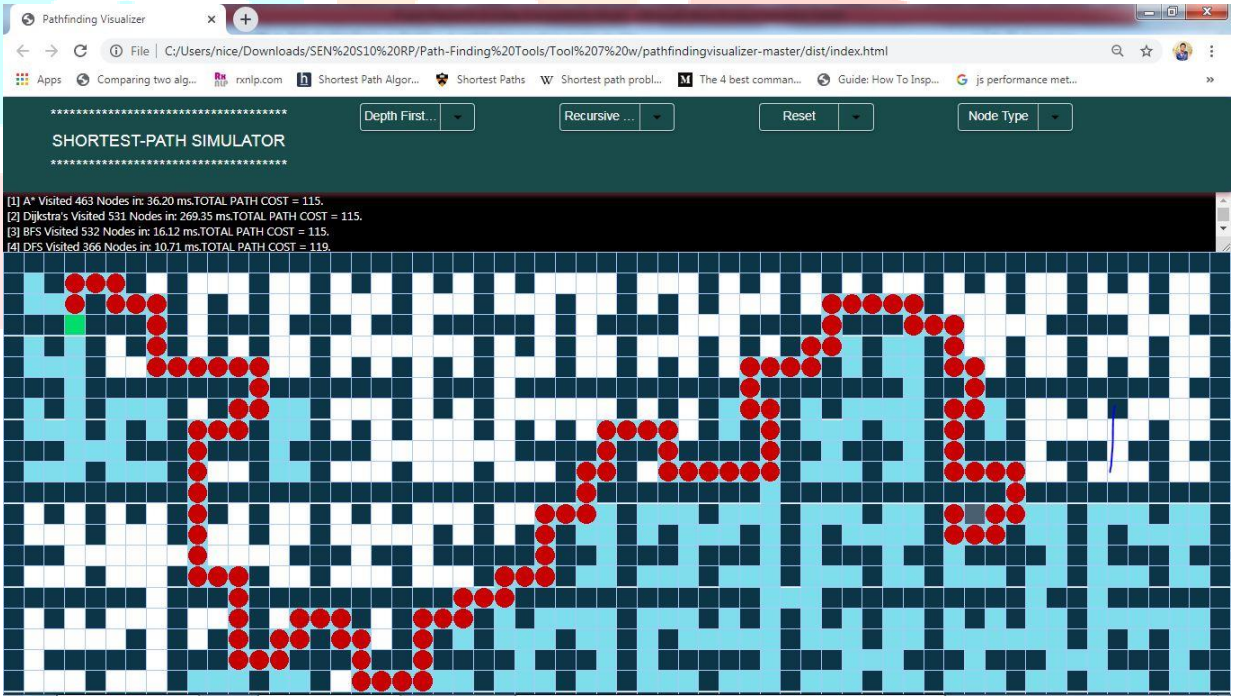


Figure 6.12: RESULT *SET OF ALL ALGORITHMS UNDER RECURSIVE TERRAIN CONDITION.*

6.1.3 RANDOM TERRAIN

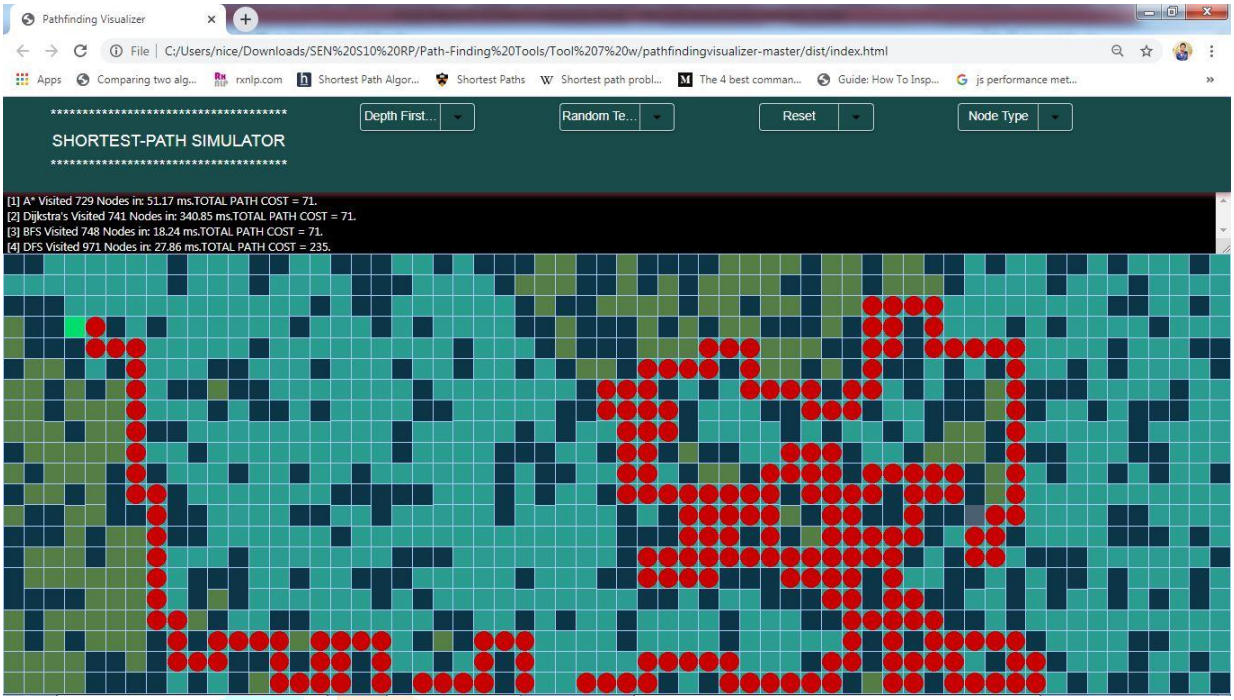


Figure 6.13: RESULT *SET OF ALL ALGORITHMS UNDER RANDOM TERRAIN CONDITION.*

6.2 CPU UTILIZATION AND MEMORY USAGE

6.2.1 A* ALGORITHM

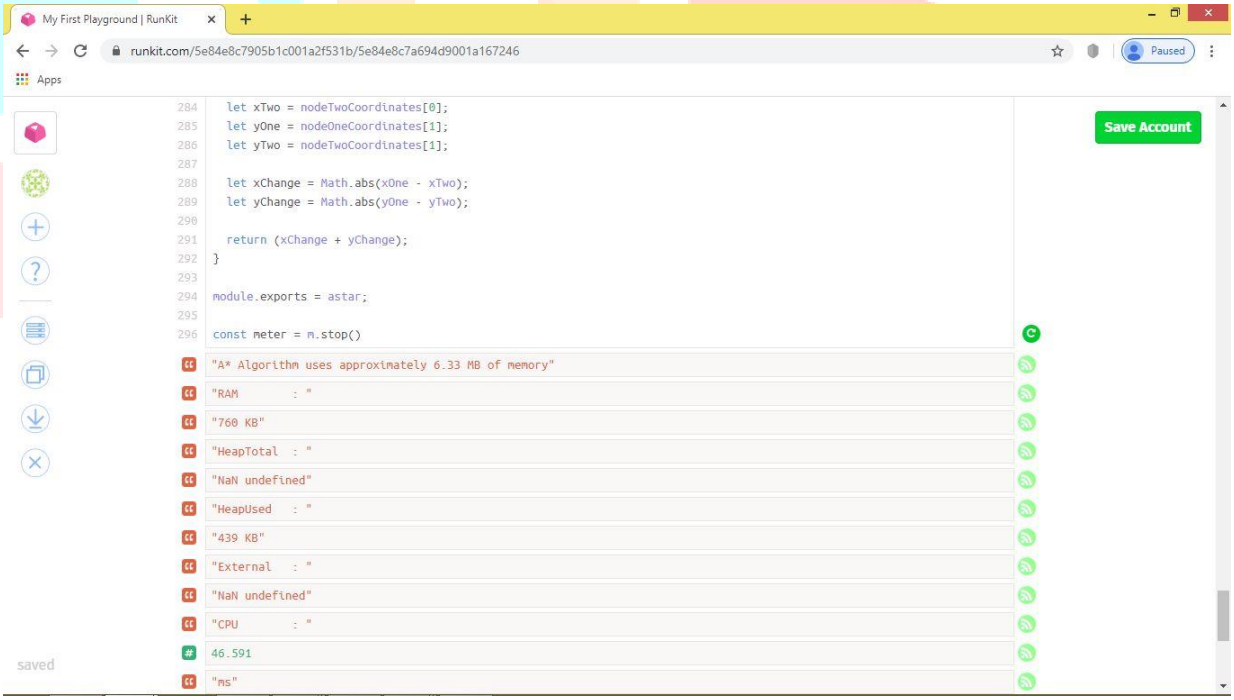


Figure **6.2.1: CPU** *USAGE OF A* CODE.*

6.2.2 DIJKSTRA ALGORITHM

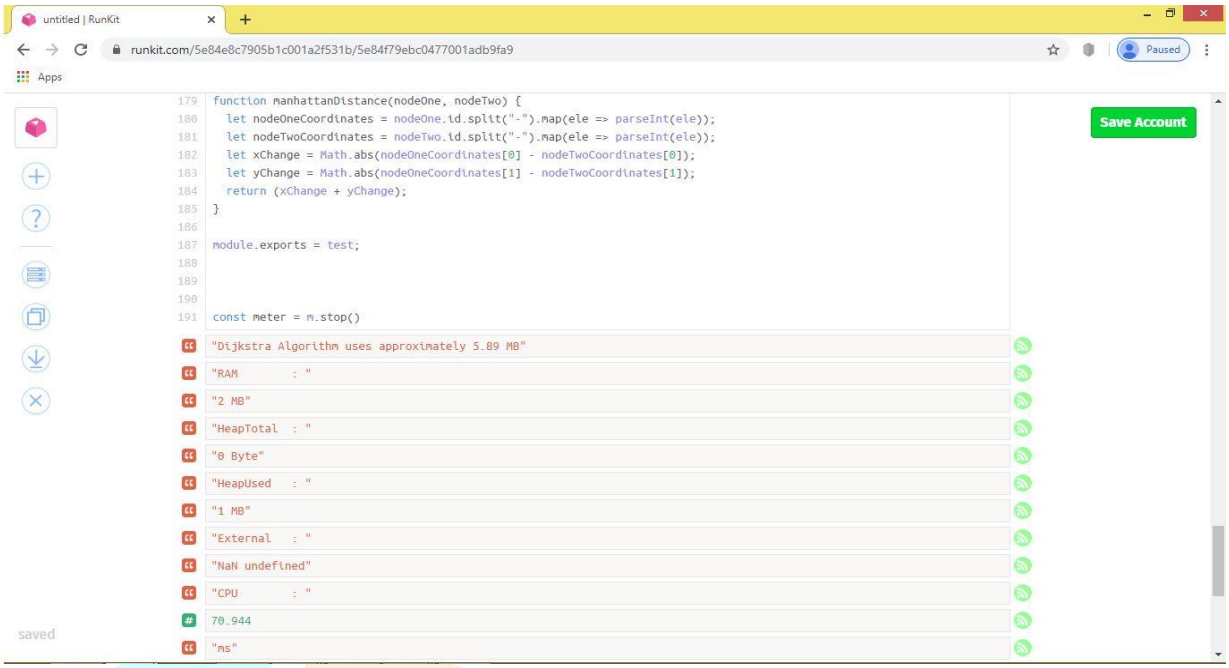


Figure **6.2.2: CPU** *USAGE OF DIJKSTRA CODE.*

6.2.3 BREADTH FIRST SEARCH ALGORITHM

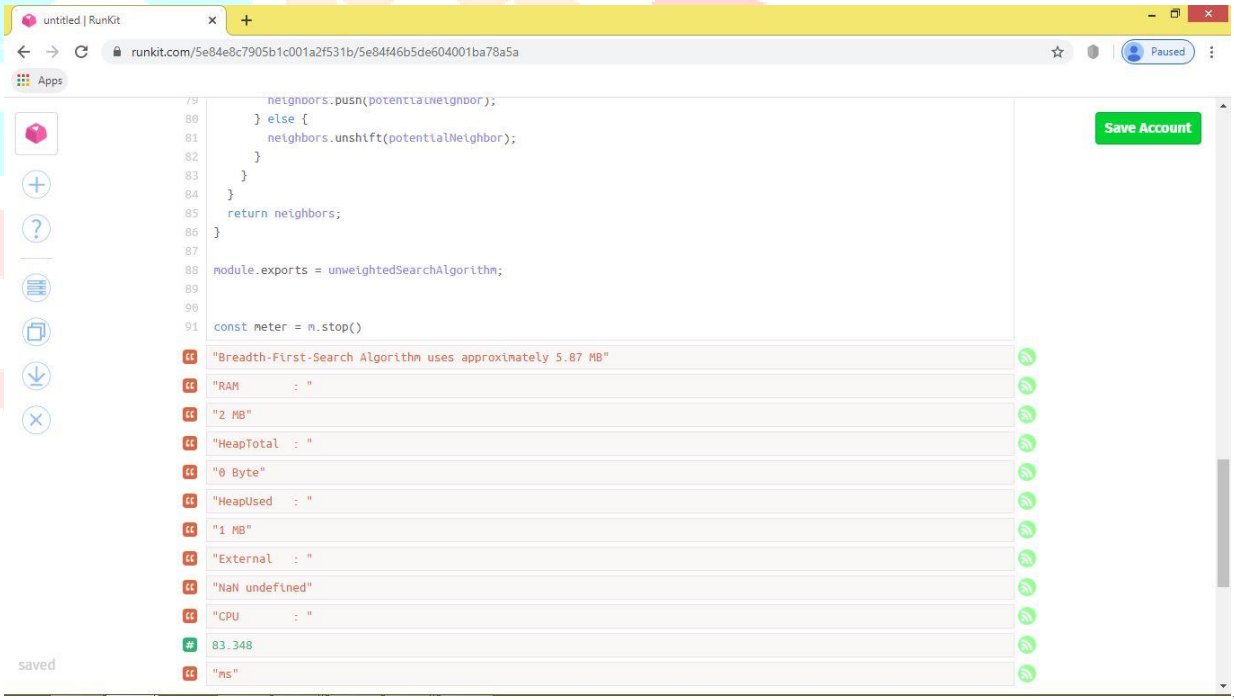


Figure **6.2.3: CPU** *USAGE OF BFS CODE.*

6.2.4 *DEPTH FIRST SEARCH ALGORITHM*

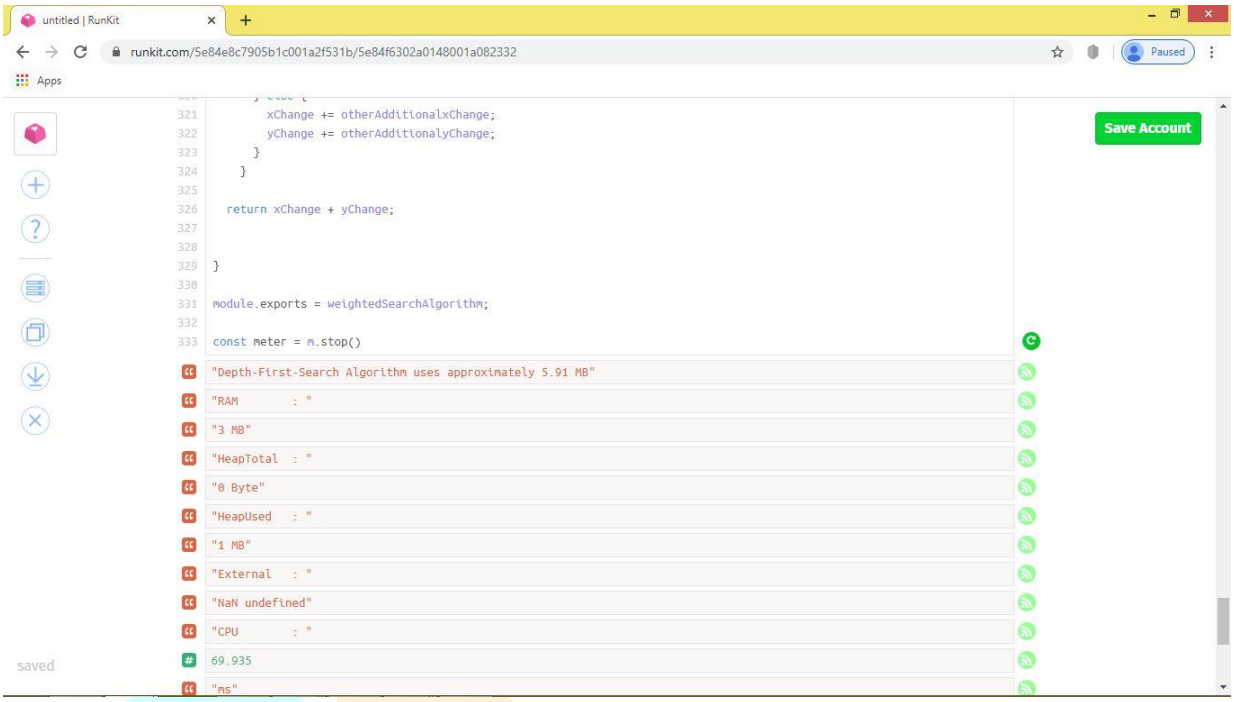


Figure **6.2.4: CPU** *USAGE OF DFS CODE*

6.3 TABULATED SUMMARY OF RESULTS

ALGORITHMS USED	TERRAIN USED	TOTAL PATH COST	TOTAL TIME TAKEN (in ms)	NO:OF NODES VISITED	TOTAL MEMORY USED (in Mb)	RAM USED (in Mb)	HEAP USED (in Kb)	CPU USAGE (in ms)
A* ALGORITHM	NORMAL	62	95.89	662	6.33	0.74	440	46.59
	RECURSIVE	115	36.20	463				
	RANDOM	71	51.17	729				
DIJKSTRA ALGORITHM	NORMAL	62	699.28	1146	5.89	2.0	1024	70.94
	RECURSIVE	115	269.35	531				
	RANDOM	71	340.85	741				
BFS ALGORITHM	NORMAL	62	31.48	1447	5.89	2.0	1024	83.35
	RECURSIVE	115	16.12	532				
	RANDOM	71	18.24	748				
DFS ALGORITHM	NORMAL	712	16.02	713	5.91	3.0	1024	71.34
	RECURSIVE	119	10.71	366				
	RANDOM	235	27.86	971				

Table 6.3: SUMMARISED RESULTS OF ALL FOUR ALGORITHMS IN A TABLURISED FORAMT.

7. FINAL VERDICT

7.1. DISCUSSIONS

The objective of this project was to analyse and compare the various shortest path algorithms in terms of their performance and complexity. The expected results of this project are to find the best algorithm that can be used for shortest path finding problem especially on a GPS system on the basis of their performance. But during the course of this project, the results obtained provided with new insights to these algorithms performance. We used mainly two platforms to bring out the best algorithm. One of them is the simulator and the other is the node js-meter. With the simulator we got the results and from those results we have calculated the average for the purpose of better graphical experience.

7.2. CONCLUSION

In this project we compared all this four algorithms fewer than seven main factors that are path cost, time taken, number of nodes visited, total memory used, RAM used, heap used and cup usage. This factors are can be prioritized like total path cost, time taken, total memory, RAM usage, cup usage, heap used and the least prioritized number of nodes visited. When we were examining the path cost factor we have observed that all the three algorithms that is Dijkstra, A*, BFS algorithms have been very efficient in finding the least cost path whereas DFS was unable to do so. So with total time taken factor Dijkstra algorithm used 19 times more seconds than BFS,A* took 3 times more than BFS, making BFS a time-friendly algorithm. When it comes to memory usage and RAM usage A* took the most out of it and rest three algorithms took less memory for usage. With the CPU usage the BFS took almost 85-90% of it whereas A* and Dijkstra were good enough. So for the concluding points, Dijkstra is known as Grand-Father of shortest-path finding algorithms as it is 64 years old but still Dijkstra managed to do well with all algorithms. BFS and DFS being good at only some un-prioritized factors needs not be considered, but the time it is taking very less than any of them. A* on the other hand does a decent job in almost all factors. So any combinations of A*, Dijkstra and BFS would benefit the process of finding the shortest path between two nodes in a shortest time.

7.3. SCOPE FOR FUTURE WORK

Searching is the problem-solving technique in computer science field. Almost all application is using Dijkstra till now. As technology grows, the speed of vehicles also increased like rocket. That is why a scholar as well as Microsoft, Google like companies starts working on A* algorithm because A* performance is better than Dijkstra. At present time, even one second has also weightage for Light motor vehicles because of its very high speed. But the major disadvantages with A* algorithm is that it takes a huge amount memory as it stores all its nodes to memory .So as both algorithms having their own dis-advantages the scope for future work is high. A study on solving this problem could also lead to new algorithms being generated from these two algorithms which can be used in all applications of computer science like computer networks, road networks and all location based services.

References

[1] Md. Firoj Ali, Rafiqul Zaman Khan,Distributed Computing: An Overview - Int. J. AdvancedNetworking and Applications Volume: 07 Issue: 01 Pages:2630-2635 (2015) ISSN: 0975-0290,June 2015.

[2] Mustafizur Rahman, Rajkumar Buyya ,Decentralization in Distributed Systems: Challenges,Technologies, and Opportunities - Published 2011 DOI:10.4018/978-1-61350-110-8.

[3] Apekshit Sharma, Chandrakant Sharma ,DISTRIBUTED COMPUTING IN MOBILE ENVIRONMENT- International Journal of Computer Science and Communication Vol. 3, No. 1, January-June 2012, pp.211-213.

[4] Nandini , The Changing Indian Telecommunication Industry - Article in SSRN Electronic Journal · December 2014.

[5] Nurul I. Sarkar, Senior Member, IEEE and Syafnidar A. Halim - A Review of Simulation of Telecommunication Networks: Simulators, Classification, Comparison, Methodologies, andRecommendations -Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), March Edition,2011.

[6] Abeer Mahmoud Garad and Mamdouth Mustafa Ismail , New Perspective of Telecommunication: A Conceptualized Framework for Teleworking – The Social Sciences 13 (4): 891-897, 2018 ISSN: 1818-5800.

[7] Boris B. Iskolnyy, Roman V. Maximov, Sergey R. Sharifullin Shtemenko Krasnodar Survivability Assessment of Distributed Information and Telecommunication Networks – CEUR-WS.org/Vol-2081/paper13.pdf.

[8] Sumitha J. - Dept. of Computer Science, Research Journal of Recent Sciences ISSN 2277-2502 Vol. 3(ISC-2013), 1-3 (2014).

[9] Dr.S.S.Dhenakaran, A.Parvathavarthini ,An Overview of Routing Protocols in Mobile Ad-Hoc Network - International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 2, February 2013.

[10] Swati Yadav ,Bahadurgarh ,Review Paper on Development of Mobile Wireless Technologies (1G to 5G) - IJCSMC, Vol. 7, Issue.5, May 2018, pg.94 – 100.

[11] Heinz Stockinger ,Distributed Database Management Systems and the Data Grid - 2001 Eighteenth IEEE Symposium on Mass.

[12] Swati Gupta Fundamental Research of Distributed Database - IJCSMS International Journal of Computer Science and Management Studies, Vol. 11, Issue 02, Aug 2011.

[13] Ning Xu ,A Survey of Sensor Network Applications - IEEE communications magazine, 2002.

[14] Stephen Naicken, Anirban Basu, Barnaby Livingston and Sethalat Rodhetbhai , A Survey of Peer-to-Peer Network Simulators - 2006 .

[15] Francis HEYLIGHEN , WorldWide Web: a distributed hypermedia paradigm for global networking - Heylighen, F. (1994). World-Wide Web: a distributed hypermedia paradigm for global networking. In Proceedings of SHARE EUROPE SPRING MEETING (pp.355–368).

[16] Uvika Kujur, Dr. Ragini Shukla , Features Analysis and Comparison of 5G Technology: A Review - International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 7, Issue 5, May 2018, ISSN: 2278 – 1323.

[17] Vasileios Karagiannis, Michael Borkowski , Edge Computing with Peer to Peer Interactions: Use Cases and Impact - Accepted at: Workshop on Fog Computing and the IoT (IoT-Fog), pp. 1–5, 2019.

[18] Pekka Pirinen , A Brief Overview of 5G Research Activities - 1st International Conference on 5G, 2014.

[19] Latesh Kumar K J , Implementing Network File System Protocol for Highly Available Clustered Applications on Network Attached Storage - 2013 5th International Conference.

[20] Md. Shohel Rana, Mohammad Khaled Sohel, and Md. Shohel Arman - Distributed Database Problems, Approaches and Solutions — A Study - International Journal of Machine Learning and Computing, Vol. 8, No. 5, October 2018.

[21] Urbano Nunes , José Alberto Fonseca , Luís Almeida , Rui Araújo and Rodrigo Maia - ,Using distributed systems in real-time control of autonomous vehicles, -Robotica (2003) volume 21, pp. 271–281. © 2003 Cambridge University Press.

[22] Carlos Eduardo Pereira, Luigi Carro, DISTRIBUTED REAL-TIME EMBEDDED SYSTEMS: RECENT ADVANCES, FUTURE TRENDS AND THEIR IMPACT ON MANUFACTURING PLANT CONTROL, May 17-19 2006.

[23] Davide Brugali, Mohamed E. Fayad ,Distributed Computing in Robotics and Automation.

[24] Cornel Klein Bernhard Rumpe,A streambased mathematical model for distributed information processing systems SysLab system model.

[25] N. Mahira Banu and S. Geetha,Industrial automation using GSM.

[26] KAZEM DASTGERD , NASSER MEHRSHAD and MOHSEN FARSHAD,A new intelligent approach for air traffic control using gravitational search algorithm, February 2016.

[27] R.E. Fields, P.C. Wright, P. Marti and M. Palmonari ,Air Traffic Control as a Distributed Cognitive System: a study of external representations.

[28] Xiao-Bing Hu, Ezequiel Di Paolo ,An efficient genetic algorithm with uniform crossover for air traffic control, 2007.

[29] Ranko Radovanović, Zaharije Radivojević, and Miloš Cvetanović ,Distributed Air Traffic Control Simulator, 2013.

[30] Raj Jain ,GPS: Applications to Distributed Systems and Networks.

[31] Rejo Mathew - Narsee,Contemporary GPS Security Mechanisms,, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-2S, December 2019.

[32] Guixia Guana, Lei Yan , Jiabin Chen , Taixia Wu,Vehicle tracking algorithm based on GPS and map-matching,July 2016.

[33] Nainesh Agarwal , Julien Basch , Paul Beckmann , Piyush Bharti , Scott Bloebaum Stefano Casadei , Andrew Chou , Per Enge , Wungkum Fong , Neesha Hathi Wallace Mann , Anant Sahai , Jesse Stone , John Tsitsiklis , Benjamin Van Roy ,Algorithms for GPS operation indoors and downtown, GPS Solutions (2002).

[34] Pablo Cotera , Miguel Velazquez , David Cruz , Luis Medina and Manuel Bandala, Indoor Robot Positioning using an Enhanced Trilateration Algorithm,, 20 March 2016.

[35] Jinhao Lu and Chi Dong, Research of Shortest Path Algorithm Based on the Data Structure, 2012.

[36] Pooja Singal - DCSA, MDU, Rohatk, R.S.Chhillar- HOD, DCSA, MDU, Rohatk - ,Dijkstra Shortest Path Algorithm using Global Positioning System, September 2014.

[37] Shrawan Kr.Sharma, B.L.Pal, Shortest Path Searching for Road Network using A* Algorithm, 7 July 2015.