iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.42720

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

A Study on Bellman Ford Algorithm for Shortest Path Detection in Global Positioning System

Avinish Rai
Birla Institute of Technology

Abstract: ***This is the Master of Technology project report of Mr. Avinish Rai (Roll No- MT/CS/45001/20), final year student in the department of Computer Science and Engineering, Birla Institute of Technology, under the supervision and guidance of Dr Ila Sahay Dayal.***

The thesis deals with Bellman ford algorithms is used to find the optimal shortest path between two or more than two nodes in the graph or in the tree. In this paper we add this bellman ford functionality in the G.P.S. this algorithm will work as the parameter in this and use from the initial node to target or source node. We will do the comparative and mathematical studies between both Dijkstra's and Bellman ford algorithm which will give optimal results.

I. INTRODUCTION

A. *Introduction to G.P.S.*

1) *Global Positioning System (GPS)*

GPS has a system of more than 30 plus navigating satellites orbiting around earth. It constantly sends out signals. It constantly sends the signals once it receives it calculate using mathematical calculation to get path.

a) Locate the position.
b) Use map to find one location to another.
c) Create digital maps.

2) *Working of G.P.S.*

Each GPS satellite send signals to allow a GPS receiver to calculate the location of the satellite. GPS receivers know this help in mathematical calculations to determine. To calculate the location of the user and show it on electronic device which is called "Trilateration". 2D position and tracking movement, the GPS unit must lock on to the radio signal on three satellites (Figure – 1).

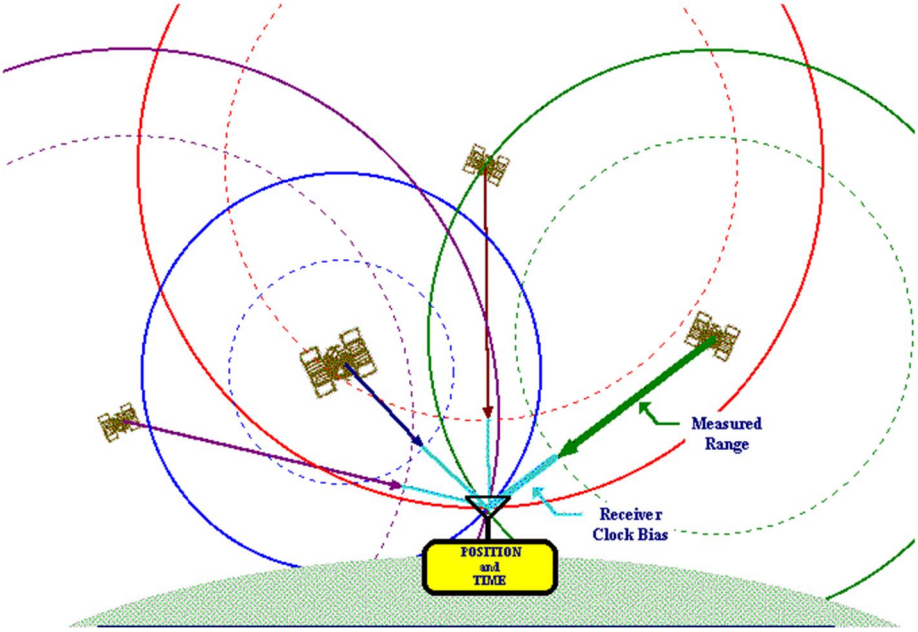


Figure – 1

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

a) Three Segments of GPS

- Satellites orbits the earth and send receive signals.
- Control by the monitoring stations.
- Used by normal public and military services where they receive signals.

As shown in Figure – 2.

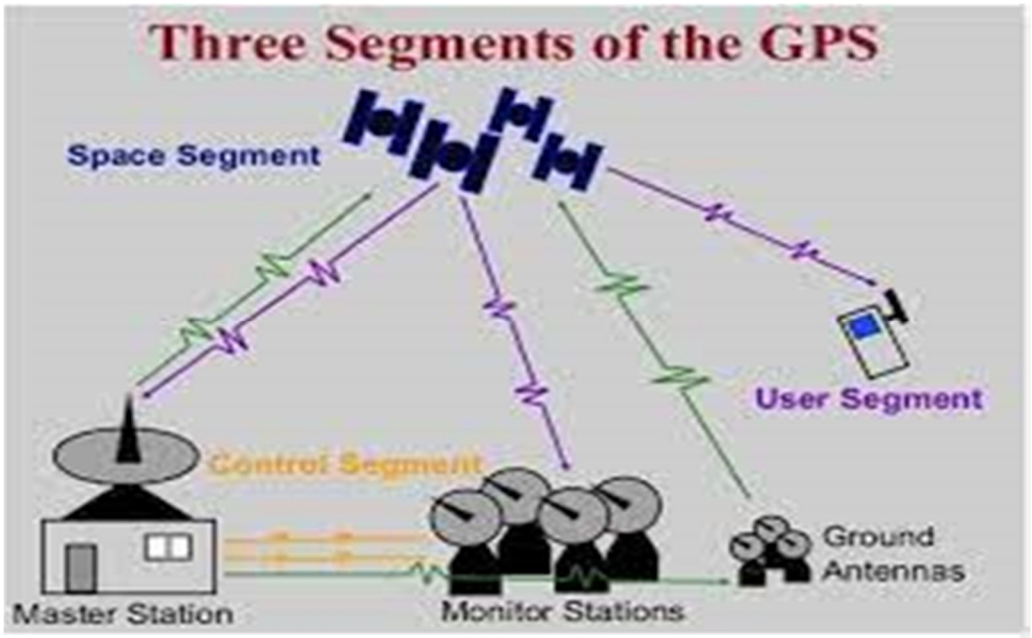


Figure – 2

3) Mathematical Algorithm followed to Find Path.

The mathematical algorithms that followed by GPS to find optimal path is Dijkstra's, but many other cabs service company also follow Dijkstra's, Nasa moon rover mission follow A* algorithms to search path in unknown environment, but the research we are doing. We will implement Bellman ford instead of Dijkstra's compare and see the results, that what difference we obtained.

II. PROBLEM DEFINITION

A. Algorithms for Shortest path Detection in G.P.S.

The short path algorithms are primarily divided in two parts. The first part is single source short path and second is called all pair short path. Single source short path is used to find out a path from single vertex to different vertex and all pair shortest path is take the smallest path from all the set of vertexes in the graph. The best is depending upon the type of a graph like how we generate shortest path algorithms to get results in the complicated graphs and bellman ford and Dijkstra's is the best algorithms to find the path. The research further decided which algorithm is better.

B. How Bellman Ford is better than Dijkstra's Algorithms.

GPS is used to track the movements and give us the accurate path or the location currently Dijkstra's algorithms are used in the G.P.S. Here we will focus on the shortest path algorithm which is bellman ford algorithms and compare with Dijkstra's algorithms and see the results. Bellman give efficient and optimal path, but Dijkstra's is good for large mesh of networks.

III. EXPERIMENTAL WORK

A. Suggested Work

This suggested work, we use in this research paper is bellman ford algorithms to find the shortest path and once we find short and optimal way from an initial node to target node, we apply Dijkstra's algorithms on same nodes and compare the output. This research work show's us the theoretical idea about. A flowchart is suggested that show the algorithm behaves.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

IV. PROPOSED MODEL FLOW CHART SHOWS HOW BELLMAN FORD WORKS IN GPS.

A flowchart has been prepared and it shows that how our algorithm work when we implement that in G.P.S. system and flowchart in Figure - 3 is given below.

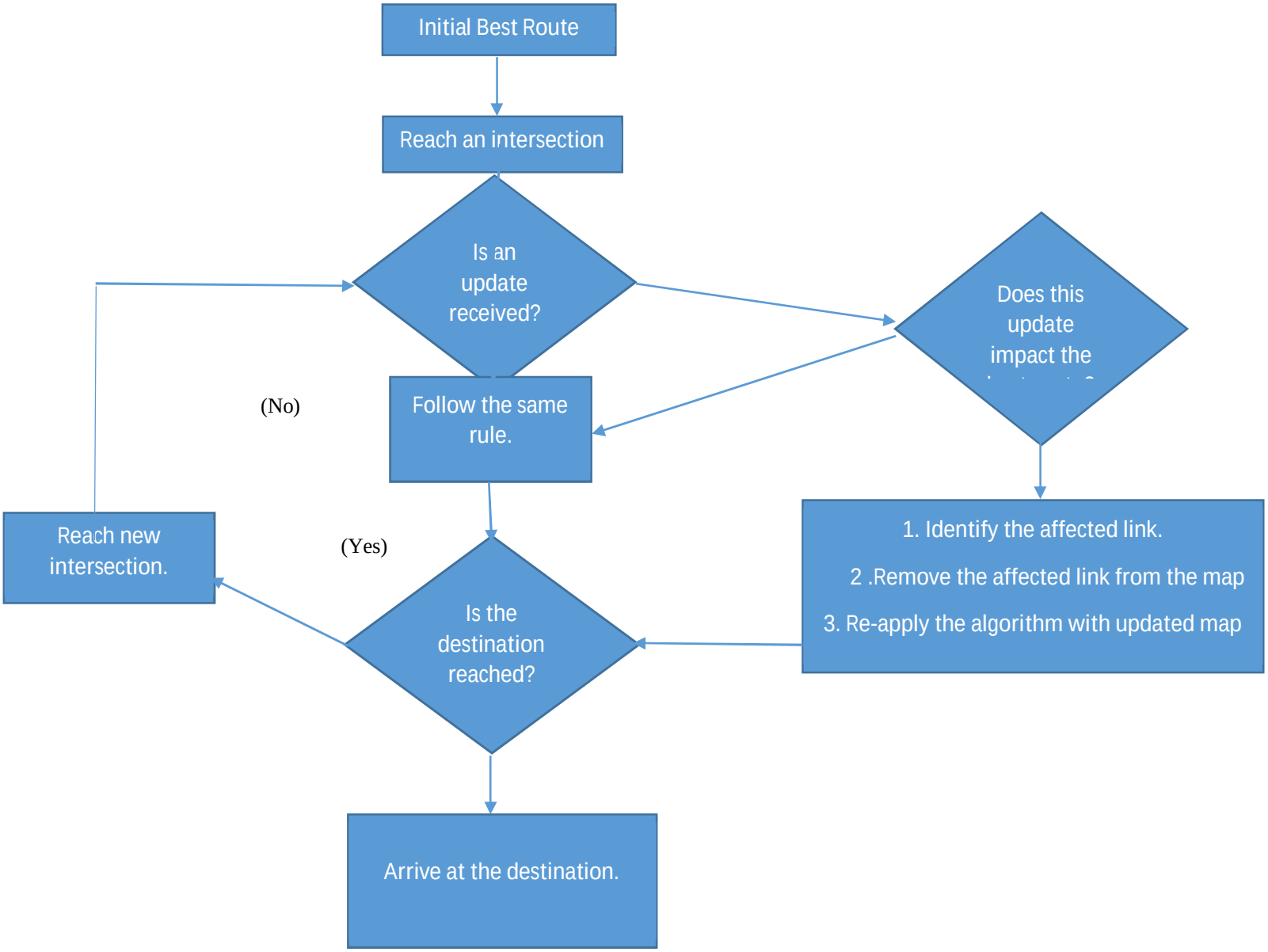


Figure – 3

V. DIJKSTRA'S ALGORITHMS

Dijkstra's algorithms were invented by the mathematician edger Dijkstra's and published in his research paper in year 1959 implemented on graph which is used to solve the single source short path problem in graph.

A. Working of DJKISTRA'S algorithm.

1) DIJKSTRA (Graph, source)
2) Make a set G
3) for single vertex i in Graph:
4) distance[i] ← INFINITY
5) previous[i] ←UNDEFINED
6) add i to G

7) distance[source] ← 0
8) while G is filled do:
9) U ←in G from main distance[x]
10) discard x from G
11) for each neighbor i of x:
12) alt ← distance [x] + length(i,x)
13) If alt< distance[i]
14) distance[i] ←alt
15) previous[i]←x
16) end while
17) return

B. Application of Dijkstra algorithms on graph.

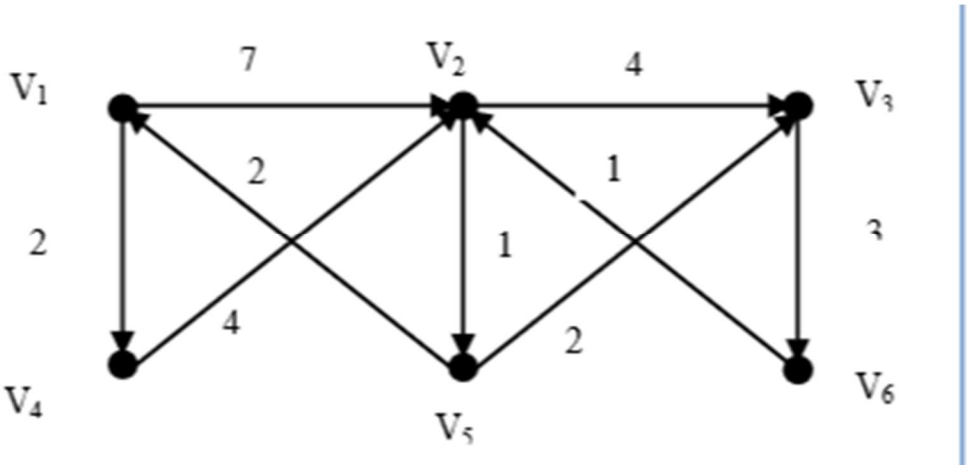


Figure – 4

Source	Node vertex 1	Node vertex 2	Node vertex 3	Node vertex 4	Node vertex 5	Node vertex 6
{}	0					
V1				2		
V1, V4		6				
V1, V4, V2					7	
V1, V4, V2, V5			9			
V1, V4, V2, V5, V3						

TOTAL DISTANCE TRAVELLED FROM V1 TO V6
V1 →2
V4→4
V2→1
V5→2
V3→3
V6.
2+4+1+2+3 = 12.

C. Time and Space Complexity Analysis for Dijkstra's.

1) (V-1) vertices are connected to each edge, neighbor edges to node is (V-1). E has (V-1) edges connected to vertex.
2) Min heap O (log (V)) + O(1) or O(log(V)) is for finding and also updating .
3) The complexity in time is E*(log V). or E*log V for all vertices of vertex which is adjacent.
4) Complexity in time for all V vertices is V * (E*log V) i. e O (V E log V). Space Complexity
5) V is the vertices in the graph O(V^V) is complexity.

D. Proof of Correctness: DIJKISTRA'S ALGORITHMS

Dijkstra's algorithms u is the vertex included in reached set, so we have.
Distance (Source, Node(u)) = distance(source, node(u)) [Distance(Source, Node(u)) distance computed by algorithm, distance(source, node(u)) is the actual distance]
Proof by contradiction
Suppose the statement is incorrect
Distance (Source , Node(u)) > distance(source, node(u))
Let x is the first vertex in the reached set, x is included in the set.

1) Distance (Source , x) > distance(source , x)

This said that all previous vertex z included in the reached set Distance(Source , z) = distance(source , z) .
P is the real path from Source to x, z is the vertex that is not in the set, y is the predecessor of x(Figure – 5)

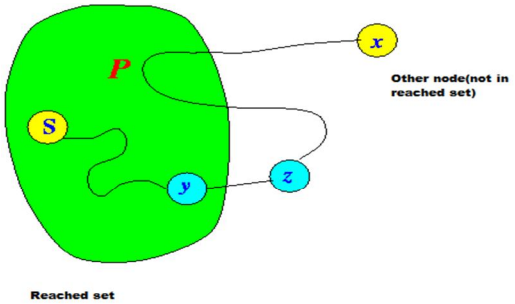


Figure – 5

- Distance (Source, y) = distance (source, y)
- Distance(Source,z) = D(Source,y) + cost(y,z) = distance(source,y) + cost(y,z)
- Distance (Source, x) ≤ D (Source, z)

2) Therefore, a sub path is the short path, so Source → x = shortest path Source → z ∪ z → x. So, distance (Source, x) = distance (Source, z) + distance (z, x). we conclude that
3) Distance (Source, x) <= D (Source, z) = d(source, y) + cost (y , z) <= distance(source, y) + cost(y, z) + d(z, x) = distance(source, x)
4) Distance (Source, x) > distance (source, x) Hence this statement is false.

VI. BELLMAN FORD ALGORITHMS

Richard Bellman invented algorithms year 1958. Bellman ford algorithms is to detect the short and optimal way as well as deals with the negative weights from source to target node. Bellman ford algorithm is the most versatile algorithms.

A. Working of Bellman ford Algorithms

1) Function Bellman Ford (0,8)
2) for each node V in G
3) distance[i]←INFINITY
4) previous[i]←NULL

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

5) distance[s]←0
6) for each node V in G
7) for each edge(i, x) in G
8) alt← distance[i]+length(i ,x)
9) if alt<distance[x]
10) distance[x]← alt
11) previous[x]←u
12) for single edge (i, x) in G
13) if distance[i] + length(i, x)<distance(x)
14) return

B. Application of Bellman ford on Graph

A graph is given with negative weights as well as positive weight apply B.F.A. in Figure – 6 .

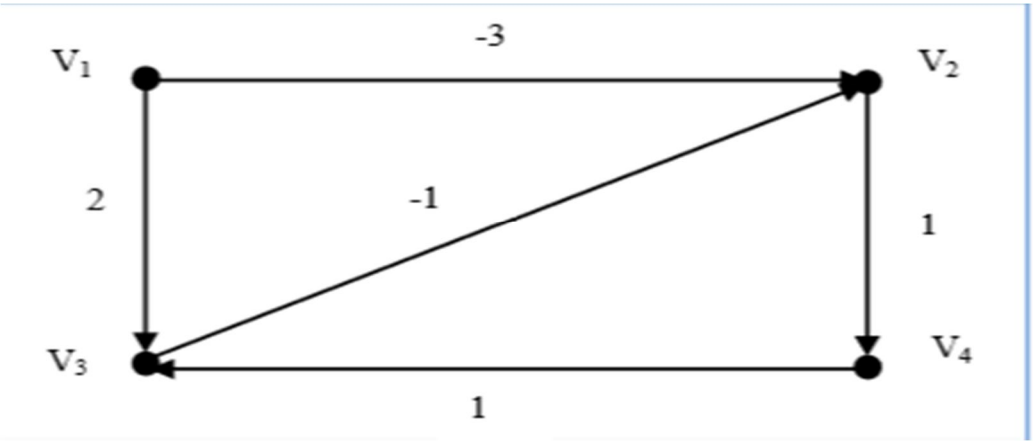


Figure – 6

Source	Vertex1	Vertex2	Vertex3	Vertex4
Vertex1	0	INFINITY	INFINITY	INFINITY
Vertex2	0	-3	2	INFINITY
Vertex3	0	-3	2	-2
Vertex4	0	-3	-1	-2

Total distance travelled from V1 to V4 is:
V1→
-3→
V2→
1→
V4
Total distance = (-3) + 1 = -2
Therefore, total distance travelled is -2.

C. Time And Space Complexity Analysis For Bellman Ford

1) Complexity in time measured for bellman ford algorithm is O(V*E).
2) V = n, E = O (n*n). So overall time complexity O (n*n*n), V is number of nodes and E is the edges.
3) Bellman ford algorithm is an algorithm that calculates the shortest path in a weighted digraph.

Space Complexity
V is the total of nodes in the graph therefore O(V) is complexity.

D. Proof of Correctness: Bellman ford algorithm.

Lemma:

After K iterations of relaxations, for any node u, dist [u] is the smallest length of a path from Source to u(node) that contains at k edges.

Proof:

Use mathematical induction

Base: after 0 iterations, all dist -values are infinity, but for dist[S] = 0, which is correct.

Induction: Proved for K → prove for K+1

Continued.

- Before K+1th iteration, dist [u] is the smallest length of a path from Source to u(node) containing at K edges.
- Each path from S to u goes through one of the incoming edge(v, u).
- Relaxing by (v, u) is comparing it with the smallest length of a path from Source to u(node) through v containing K+1 edges.

Corollary 1: (With positive cycle)

- In a graph without negative weight cycles, Bellman-Ford algorithm correctly find all distances from starting node S.

Corollary 2: (With negative cycle)

- Therefore is no negative cycle reachable from Source such that u is reachable from this negative weight cycle, Bellman ford correctly find dist. [u] = d(S, u).

VII. ADVANTAGE OF BELLMAN FORD ALGORITHMS OVER DJKISTRA'S ALGORITHMS.

1) Bellman-Ford is a single-source shortest path algorithms same as Dijkstra but deal with negative weights.
2) The main difference between the algorithms is that Bellman-Ford is well versed of handling negative costs.
3) Dijkstra's algorithm greedily selects the minimum-weight node.
4) Bellman ford is also simpler than Dijkstra and suites well for distributed systems.
5) Bellman–Ford relaxes all the edges, and does this |V | − 1 times, where |V | is the number of vertices in the graph.

VIII. PROPOSED ALGORITHMS FOR SHORTEST PATH DETECTION USING BELL MAN FORD ALGORITHMS IN G.P.S.

1) Single Source point(H, s)
2) for i ← 1 to |v[H]| - 1
3) do for each edge (z , x) = E[G]
4) do RELAX (z, x, y)
5) for each edge (z, x) = E[H]
6) d[x] > d[z] + w(z, x)
7) then return false
8) Return true.

IX. APPLY BELLMAN FORD ALGORITHMS AND DIJKSTRA ALGORITHM MATHEMATICALLY ON SAME GRAPH.

Example on shortest path.

A Figure – 7 is given below, have to apply both algorithms mathematically and analysis their respective results.

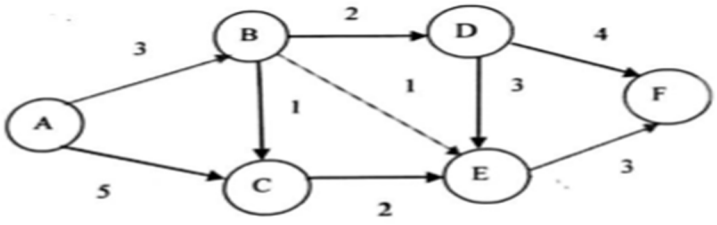


Figure – 7

A. Applying Dijkstra Algorithm

The table below shows the algorithms travelled from source to target node using in Dijkstra's algorithms.

Number of steps.	N (path travelled)	Position from source to target	Distance	Path to destination	Path to destination	Path to destination	Path to destination	Path to destination
1	a	2→3	0→0		5(a→c)			
2	a→b	5→7	0→5	3(a→b)				
3	a→b→c	8→11	0→10		4(a→b→c)			
4	a→b→c→e	11→15	0→15			5,a→b→d		
5	a→b→c→d→e	14→19	0→20				4(a→b→e)	
6	a→b→c→d→e	17→23	0→25					7(a→b→e→f)

Node Covered = a (initial node) → b → c →d→e→f (target node)
Total Distance covered = 3+1+1+1+2+3+4 = 15
Cost to travel from initial to target node = a → 3 →b → 4→ e→ 7.

B. Applying Bellman ford Algorithm

The table below shows the algorithms travelled from source to target node using in Bellman ford algorithms.

Number of steps	N (path travelled)	Position from source to target	Distance	Path to destination	Path to destination	Path to destination	Path to destination	Path to destination
1	a	2→3	0→0		4(a→c)			
2	a→b	5→7	0→5	3(a→b)				
3	a→b→c	8→11	0→10		4(a→b→c)			
4	a→b→c→e	11→15	0→15			5(a→b→d)		
5	a→b→c→d→e	14→19	0→20				4(a→b→e)	
6	a→b→c→d→e	17→23	0→25					7(a→b→e→f)

Node Covered = A (Initial)→B→D→E→F(Target)
Total Distance covered = 3+2+1+4 = 10. Cost to travel from initial to target = A → 3 →B → 4→ E→ 7.

X. RESULT AND ANALYSIS

Result of Experiment conducted and its analysis.

The comparative experiment done between both bellman ford and Dijkstra's algorithms, which show that which algorithms works better to detect shortest path.

Algorithm	Distance Covered	Node Covered	Cost
Dijkstra Algorithm	15	6	7
Bellman ford Algorithm	10	5	7

As you can see the above result, we concluded that if we apply Bellman ford algorithm on the global positioning system software it can give us the optimal path and without travelling the all possible shortest path it cover only the main node and it can use the negative path and search for optimal path to cover small distance and cover less node and give the shortest path but the cost will be same for both algorithm. This bellman ford visit the vertex more than once but the Dijkstra Algorithms don't, so that's why it's give us optimal and correct distance between initial node to target node.

XI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

In this paper, Bellman Ford Algorithm is implemented as optimal shortest path detection in example using the graph and then compared with Dijkstra's algorithms, which give efficient and optimal path in G.P.S. Implementation of this algorithms in G.P.S. is not done in this paper only analysis is done between both algorithm but in future can use these algorithms there are many more optimal and best concept like A* and war shall algorithms but that will be another part of research.

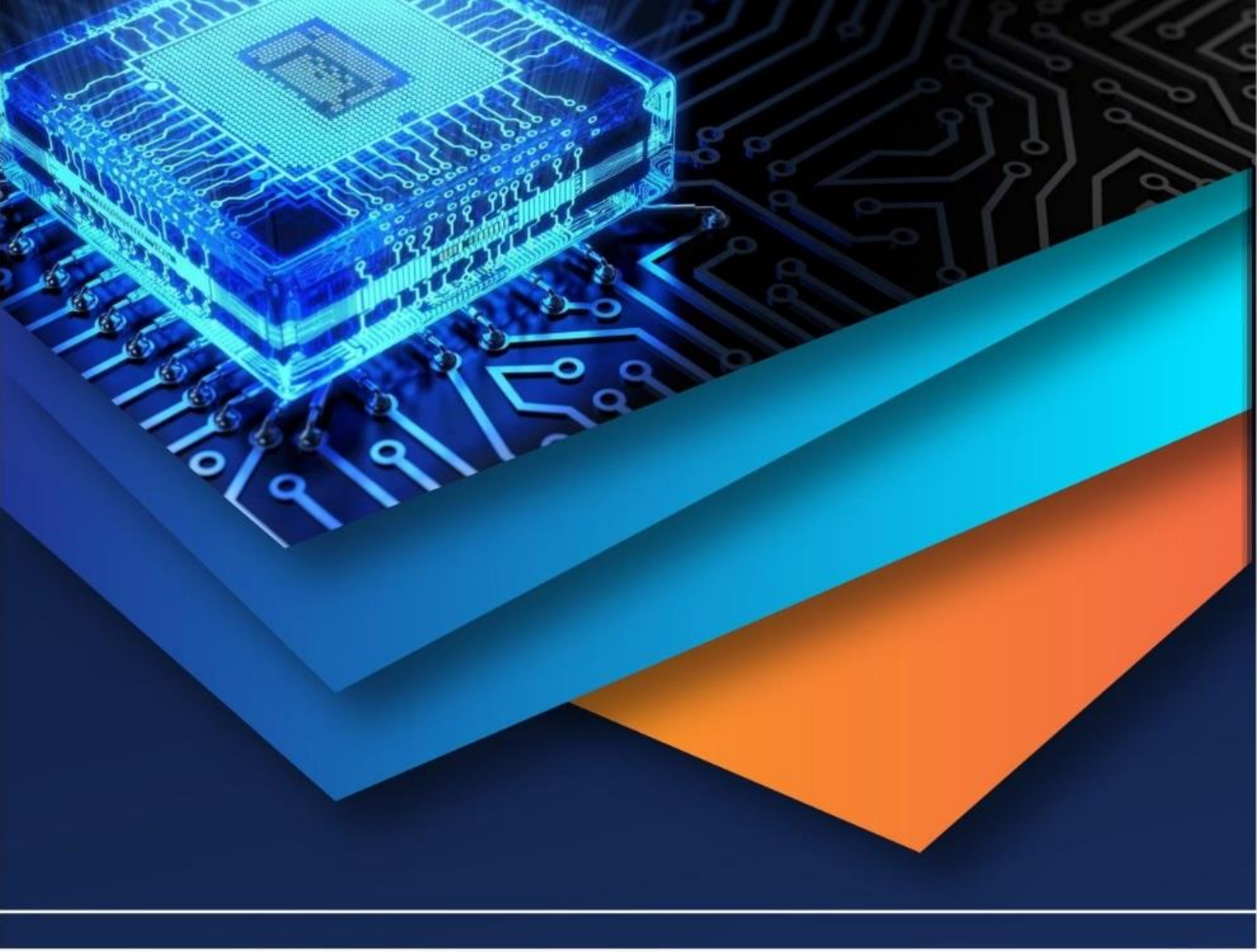
B. Future Scope

The above results give us the optimal distance, but the complexity is not better than Dijkstra's, many techs giant company like Uber, Ola, Quick ride, Tesla etc. modify and develop these new algorithms with existing algorithms in this paper try to develop algorithm based on bellman ford which is like it. But all these algorithms are built and implemented in the known environment where we can get the map from the satellite, but what will happen when we are in unknow environment how can we get the optimal path, but this research are further part of the artificial intelligence Cornell university professor and research scientist develop Physical-A* algorithm (PHA*) to detect path in other unknown environments.

BIBLIOGRAPHY

[1] Pooja Singal, R.S. Chillar "Dijkstra shortest path in global positioning system", International Journal of Computer application volume -101-No.6, September 2014, https://www.ijcaonline.org.
[2] MS Kalpana, Mr. Abhishek Tyagi "Bellman ford shortest path in global positioning system", International Research Journal of Engineering and Technology volume – 4, issue – 4|Apr-2017 www.irjet.net.
[3] Graph Theory by Narsingh Deo, mathematical application of Dijkstra and bellman ford in Graph Theory, Chapter -11 Graph theoretic algorithm and computer program, Dover Publication – 2016.
[4] Mathematics for Computer science by Tom Leighton Lecture-6 Graph theory and coloring, MIT Open courseware, issue – Fall 2010, ocw.mit.edu.
[5] Introduction to Data structure and Algorithms by Dr Naveen Garg (IIT DELHI), Lecture-7 Trees walks, NPTEL, Date – 24 /sept /2008, https://nptel.ac.in
[6] Jitendra Bahadur Singh, RC Tripathi "Investigation of Bellman-Ford Algorithms, Dijkstra Algorithms for suitability of SPP", Volume-6, Issue 1, Date – 2018, www.ijedr.org.
[7] Design and analysis of algorithms by Madhav Mukund (Chennai Mathematical Institute), Lecture- 25, 26, NPTEL. Date 26 August 2019 http://onlinecourses.nptel.ac.in/noc19_cs47/preview.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)