© 2018 IJEDR | Volume 6, Issue 1 | ISSN: 2321-9939

Investigation of Bellman–Ford Algorithm, Dijkstra's Algorithm for suitability of SPP

Jitendra Bahadur Singh[1], R.C.Tripathi[2]
Electronics Engineering Dept.,NGBU, Allahabad (India) [1]
Dean Research, NGBU, Allahabad (India) [2]

Abstract: For graph edges (weights or distance), source node are defined. Shortest path problems solve by the Algorithms, are called shortest path algorithms in this edges are labeled by a positive real number. Shortest path problems are related with either paths between a source node and destination node i.e. single source shortest path or paths among all pairs of stations i.e. all pairs shortest path. In this paper Bellman–Ford algorithm and Dijkstra's algorithm are discussed and compared the results for small no. of nodes as well as for large no. of nodes. The investigation helps to identify and suggest that which algorithm is used for a particular variant in the shortest path problems.

Keywords: WSN, Bellman–Ford algorithm, Dijkstra's algorithm, SPP, Run Time

---

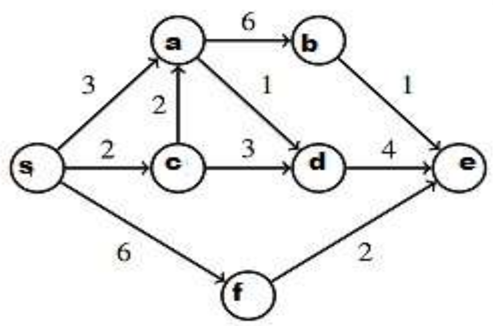
I. Introduction

In present scenario fast data transmission is essential requirements. A WSN has thousands no of sensors nodes which link with each other or to an outside destination sensor node. Shortest path algorithms are used to search directions among physical locations. In this problems are defined for directed, undirected or mixed graphs. The single-source shortest path problem solves by using Dijkstra's algorithm and Bellman–Ford algorithm and A* search algorithm while all pairs shortest paths solve by Floyd–Warshall algorithm and Johnson's algorithm. Assume w is a weighted graph, u and v are two nodes on G, E is a path in G from u and v then the weight of the path w is minimum weight of v from u via w i.e. sum of the edges weights minimum on the w. In shortest path problem, path is from a given node (source node) to all other nodes in the graph. [1-4].

II. Literature review

Dijkstra's Algorithm

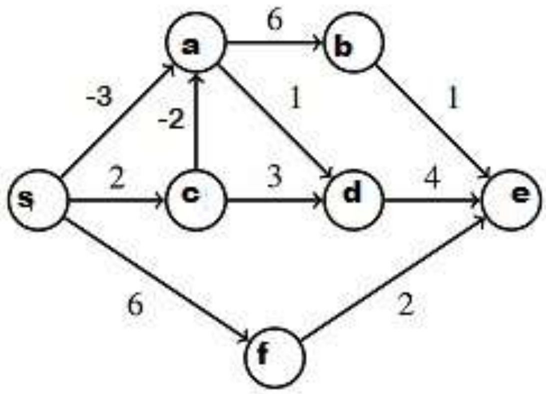
It was found by scientist Dijkstra in 1956 and in 1959 it was published [2,3].This is used for solve the weighted graph problem SSSP(Single Source Shortest Path) and producing a SP tree. The node search process continues by the algorithm and consider it next node as source node and search ends (terminates) at the destination. It is an example of efficient greedy algorithm. In this a path never shorter by visiting a node twice because of the edges have no negative weights. For SP visit the simple paths only. In SP the sub-Paths property is a key property. This algorithm search a SP tree from a source node (single) with making a set of nodes that have min. distance from source node



The graph has the nodes u or v as in the algorithm, two nodes (u, v) connected by an edge that has weight w (u, v). With initializing three values as (i) dist, an array of distances from the source node s to each node, initialized the as dist(s) = 0; and for all other nodes v, dist (v) =∞. This is done at the beginning. (ii) Q, a queue of all nodes and at the end (algorithm's progress) Q will be empty. (iii) S, indicate which nodes visited by the algorithm and at the end (algorithm's run), it contains all the nodes of the graph. This algorithm has advantages (i) Greedy Algorithm (ii) Directed & undirected Graphs. Disadvantages are (i) blind search (ii) Negative edges not handle (iii) Leads to acyclic graphs. Applications are (i) Google Maps (ii) Geographical Maps (iii) IP routing to find open SP first (iv) Telephone n/w.

Bellman Ford Algorithm

This algorithm was published in 1958 by Richard Bellman and in 1956 by Lester Ford [11].This is a dynamic programming algo. It searches the structure of the graph and generate the better solution. This algorithm applied to search the SP where negative weighted may be in graph from source node to all other nodes. Dijkstra's Algorithm is faster than Bellman Ford Algorithm but Bellman Ford Algorithm is more versatile. This Algo has the principal of relaxation



This algorithm has the advantages (i) Cost minimization (ii) Maximize the performance (iii) It allows splitting traffic, disadvantages (i) In Routing Information Protocol weightings aren't into consideration (ii) Slow response to change in the n/w topology applications are (i) . used in distance- vector routing protocols (ii) Saving network resource (iii) Routing Information Protocol (RIP).[5-10][12]

III. Time and Space complexity

Let a graph[G] with the vertices or nodes [V] and the edges[E]

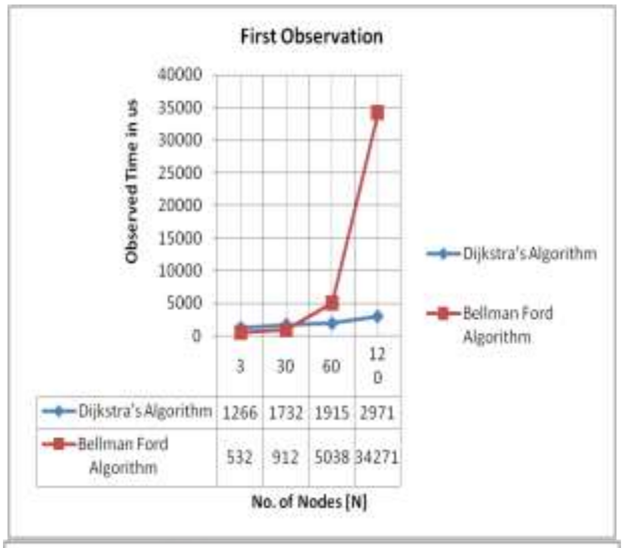
Algorithm	Time Complexity	Space Complexity
Dijkstra	O(|E|+V|Log|V|)	$O(V^2)$
Bellman-Ford	O(VE)	$O(V^2)$

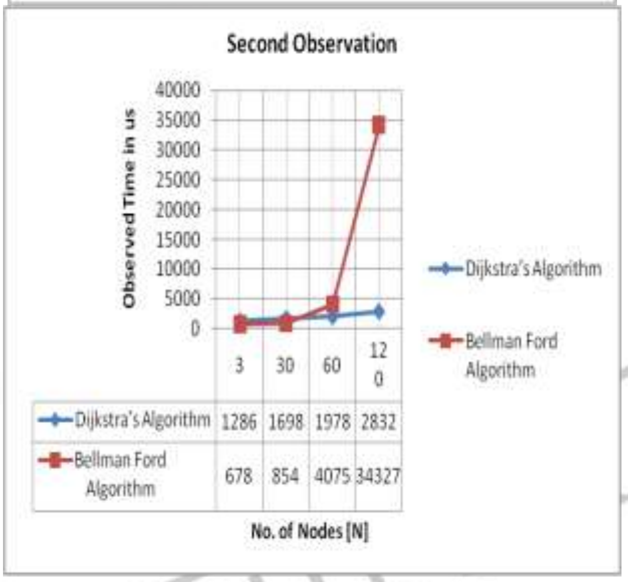
IV. Dijkstra's Algorithm and Bellman Ford's Algorithm Structure

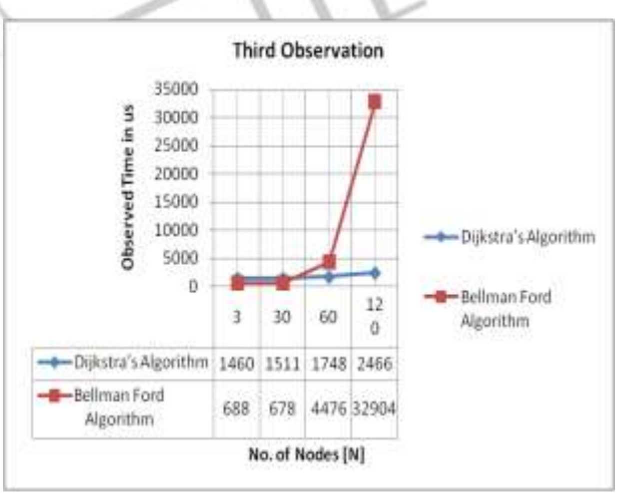
```
function dijkstra(G, S)
    for each vertex V in G
        distance[V] <- infinite
        previous[V] <- NULL
        If V != S, add V to Priority Queue Q

    distance[S] <- 0

    while Q IS NOT EMPTY
        U <- Extract MIN from Q
        for each unvisited neighbour V of U
            tempDistance <- distance[U] + edge_weight(U, V)
            if tempDistance < distance[V]
                distance[V] <- tempDistance
                previous[V] <- U

    return distance[], previous[]

function bellmanFord(G, S)
    for each vertex V in G
        distance[V] <- infinite
        previous[V] <- NULL


    distance[S] <- 0

    for each vertex V in G

        for each edge (U,V) in G
            tempDistance <- distance[U] + edge_weight(U, V)
            if tempDistance < distance[V]
                distance[V] <- tempDistance
                previous[V] <- U

    for each edge (U,V) in G
        If distance[U] + edge_weight(U, V) < distance[V}
            Error: Negative Cycle Exists

    return distance[], previous[]
```

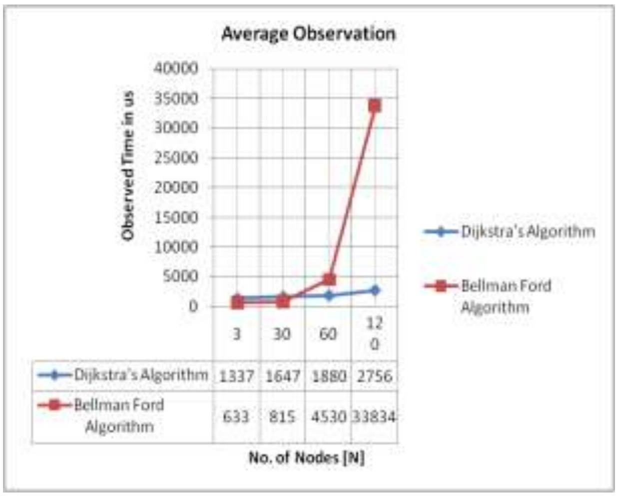
V. COMPARISON

We observe the efficiency of SP algorithm by analysis the running time of algorithms Dijkstra's and Bellman Ford. Observe running time for both algorithms Dijkstra's and Bellman Ford is in micro seconds. No. of nodes select for each observation are on random basis for a graph [G].









VI. Conclusion

In this paper authors observe the running time of algorithms Dijkstra's and in micro seconds and compare the observations with average running time and conclude that Bellman Ford algorithm is more efficient than Dijkstra's algorithm for small no of nodes, while Dijkstra's algorithm is more efficient for large no. of nodes as compare to Bellman Ford algorithm for SP problem.

ACKNOWLEDGEMENTS

The authors would like to thank the management and supporting staff of NGBU, Allahabad, India for providing support. This research work has been carried out in Research Centre of NGBU, Allahabad, India.

REFERENCES

1. T. Li, L. Qi, and D. Ruan, ―An Efficient Algorithm for the Single-Source Shortest Path Problem in Graph Theory‖, Proc. of 3rd International Conference on Intelligent System and Knowledge Engineering, Vol. 1, pp. 152-157, 2008

2. Peter Hofner, and Bernhard Moller. 2012." Dijkstra, Floyd and Warshall meet Kleene". Formal Aspects of Computing, 459–476.

3. Huang, B, Wu, Q, and Zhan, F.B. 2007. "A shortest path algorithm with novel heuristics for dynamic transportation networks. International Journal of Geographical Information Science. Vol. 21, No. 6, 625–644.

4. Asmaa Shaker Ashoor.2015. "Performance Analysis between Distance Vector Algorithm (DVA) & Link State Algorithm (LSA) For Routing Network".International Journal of Scientific & Technology Research. Vol 4, ISSN 02,101- 105.

5. Anu Pradhan, and Kumar, G. 2013. "Finding All-Pairs Shortest Path for a Large-Scale Transportation Network Using Parallel Floyd-Warshall and Parallel Dijkstra Algorithm". Journal of Computing in Civil Engineering ASCE. 263-273.

6. Forouzan, B. A. "TCP/IP Protocol Suite". 2010. Prentice Hall. ISBN-13: 978-0-13-2126953. 282-333.

7. Wei Zhang , Hao Chen , Chong Jiang , Lin Zhu. 2013. "Improvement And Experimental Evaluation Bellman-Ford Algorithm". International Conference on Advanced Information and Communication Technology for Education ICAICTE, 150-153.

8. Hemalatha, S, and Valsalal, P. 2012. "Identification of Optimal Path in Power System Network Using Bellman Ford AlgorithmS". Hindawi Publishing Corporation Modelling and Simulation in Engineering.

9. Andrew, S, and David, J. 2011. "Computer network". ISBN-13: 978-0-13-212695-3. 350-366.

10. Mariusz Głabowski, Bartosz Musznicki, Przemysław Nowak , and Piotr Zwierzykowski. 2013. "Efficiency Evaluation of Shortest Path Algorithms". The Ninth Advanced International Conference on Telecommunications (IARIA). ISBN: 978-1-61208-279-0.

11. Kevin,R. Hutson, Terri L, Schlosser, and Douglas, R. Shier.2007. "On the Distributed Bellman-Ford Algorithm and the Looping Problem" .Informs Journal on Computing. Vol 19, No 4, 542-551.

12. Cormen, Thomas H, Leiserson, Charles E, and Rivest, Ronald L.Stein Clifford . 2001. "Johnson's algorithm for sparse graphs". Introduction to Algorithms. ISBN 978-0-262 03293-3. 636–640.