# A survey paper of Bellman-ford algorithm and Dijkstra algorithm for finding shortest path in GIS application

Vaibhavi Patel[#1], Prof.ChitraBaggar[#2]
[#1]ME in Information Technology, KalolInstitute of Technology & research Center, Gujarat, India
[#2] Assistant Professor, KalolInstitute of Technology & research Center, Gujarat, India

***Abstract*: -** GIS application is very useful in transportation management. A Road network analysis is an important function of GIS, and the shortest path analysis is the key issue of network analyses. In Most of GIS application, Dijkstra algorithm is useful. . The functionality of Dijkstra's original algorithm can be extended with a variety of modifications. This paper represents the survey of two different algorithms dijkstra algorithm and bellman-ford algorithm. The purpose of the paper is to select one best algorithm from the two and. This algorithm can be used in government sector, emergency system, Business sector etc.

***Keywords*-** GIS, Optimal path, Dijkstra algorithm, Bellman-ford algorithm…

## I INTRODUCTION

A geographic information system (GIS) is a computer-based tool. Using these tools we can create, manipulate, and analyze, store and display spatial information. Spatial information is information about objects which are on earth like city, railway track, river etc… GIS tools are used in planning, administration, analyzing in different countries. GIS s widely used in government agencies, transportation agencies, emergency systems etc…

Network analysis is important function in GIS which is include shortest path problem, nearest neighbor query, resource allocation etc… A shortest path is critical problem in GIS. In many GIS applications, on the shortest path analysis of real-time requirements are also higher, such as 111, 108 fire and medical care system [1].

Shortest path problem is major problem in transportation management. Role of GIS in transportation management is very important. And routes planning are main problem in transportation agencies. IF you want to find shortest path, you have to have information about source and destination station.

The existing shortest path algorithms are many but this paper represent survey of two different shortest path algorithm dijkstra algorithm and bellman-ford algorithm used in GIS. And also present comparison based on time complexity and space complicity.

A road network can be considered as a graph with positive weights. The nodes represent road junctions and each edge of the graph is associated with a road segment between two junctions. The weight of an edge may correspond to the length of the associated road segment, the time needed to traverse the segment or the cost of traversing the segment [6].

## II. BELLMAN-FORD ALGORITHM

The Bellman-Ford algorithm uses relaxation to find single source shortest paths on directed graphs. And it is also contain negative edges. The algorithm will also detect if there are any negative weight cycles (such that there is no solution). If talking about distances on a map, there is no any negative distance. The basic structure of bellman-ford algorithm is similar to dijkstra algorithm. It relaxes all the edges, and does this |V| - 1 time, where |V| is the number of vertices in the graph [2]. The cost of a path is the sum of edge weights in the path. Now srv is source and this is single source shortest path problem so find distance form single source to different destination. This algorithm return value that negative cycle is present or not and also return shortest path. This algorithm find shortest path in bottom up manner [3].

### *a. Steps for algorithm*

This implementation is representing in graph. We have given a weighted, directed graph **G = (V, E)**. Hear "**V**" is represent as list of vertices and "**E**" is represent as list of edges. Vertex is represent as "**v**" and edges is represent as "**e**" and source vertex is represent as "**srv**" and also given weighted function **w**.

**1 initialize graph**
Bellman-ford (vertices V, edges E, source vertex srv)
For each **v** in **V**

```
{
If   v = srvthan d[v] =0
Else d[v] = infinite
Cost[v] = null
}
```

**2. Relaxation of edges**

```
For i form 1 to IVI-1
{
 For each [u,v] with w  in E
  {
    If d[u]+w<d[v]
      {
        If d[v]=d[u]+w
           Cost[v] =u
        }
      }
  }
```

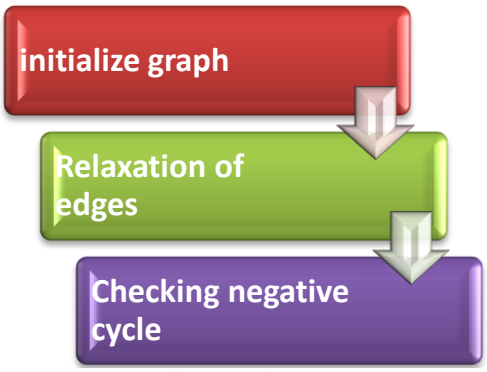
**3. Checking negative cycle**

```
For each e(u ,v) with weight  winE
{
  If d[u]+w,d[v]
    {
       Print "graph has negative cycle"
    }
}
```

*b. How to work…*

First of all we have to insert input as graph and a source vertex srv. After completion of this process we will get output as shortest path from single source to all vertices and if graph contain negative cycle then it detect it.

First step is initializes graph in which we initialize distances from source to all vertices is infinite and distance to source as 0.




Fig, 1 Process of Bellman ford algorithm

Create an array d [] for distance. After initialization we have to find shortest distance still IVI-1. Last step check if negative cycle is present or not?

## III. DIJKSTRA'S ALGORITHM

Dijkstra algorithm is also known as single source shortest path algorithm. It calculates the shortest path from the source to each of the un visited vertices in the graph [1]. Dijkstra Algorithm used the method of increasing node by node to get a shortest path tree which makes the starting point as its root. Here is the concrete method: in the Weighted directed graph, the shortest path node whichStarts from the starting point s and reaches the earliest must be the smallest point where all the nodes adjacent to s and its length of arc is Chord length [4]. if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities

*a. Steps for algorithm*

This implementation is representing in graph. We have given a weighted, directed graph **G = (V, E)** with source vertex **s.** S is represent as visited vertices. And priority queue Q is represent as all set of vertices in graph

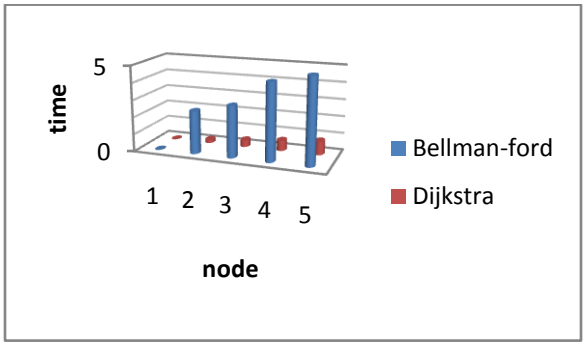
```
Dijkstra (G,s)
For each vertex v in graph G
 {
   D[s] = 0
   D[v] = ∞
 }
Initialize visited vertices S in graph G
S = null
Initialize Queue Q as set of all nodes in graph G
Q = all vertices V
While   Q ≠ ∅
{
u=mind (Graph G, distance d)
   Visited vertices S = S+u
for each vertex v in neighbor[u]
  {
      If d[v] > d[u] + w(u,v)
          Then d[v] = d[u] + w (u,v)
   }
Return d
}
```

*b. How to work…*

First of all we have initialize the graph G with list of vertices V end list of edges E. then next assign distance for every node and set 0 value to current node and set ∞ to another unvisited node. Then calculate distance from current node to its entire neighbor node and marked as visited node. Than visited node cannot be used again. Its distance store and now it is final and minimal. Current node and. Set unmarked node with less distance as current node and then continue from previous step

## IV COMPARISON OF THIS TWO ALGORITHM BASED ON TIME COMPLEXITY AND SPACE COMPLEXITY

A time taken by of Dijkstra's algorithm on a graph with edges E and vertices V can be expressed as a function of |E| and |V| using the Big-O notation.




Fig, 2 Graph of time complexity

The simplest implementation of the Dijkstra's algorithm stores vertices in Linked list or array, a time taken by algorithm is $0(|V|^2)$. And time taken by bellman ford algorithm is O(| V | · | E |). So we can clearly show that bellman ford algorithm takes more time then dijkstra algorithm. So running time of bellman ford algorithm is more than dijkstra algorithm [5].

Implement Q using priority queue At most E edges in the heap. Space complexity of dijkstra algorithm is O (V+E). And space complexity of bellman ford algorithm is O(V). So bellman ford algorithm takes more space than dijkstra algorithm.

Table-1

Time complexity

| | | **Negative edges?** | **Negative cycle?** | **Time complexity** |
|---|---|---|---|---|
| **Dijkstra** | Single source | No | No | $0(\|V\|^2)$ |
| **Bellman-ford** | Single source | Yes | Yes | O(\| V \| · \| E \|) |

According above study dijkstra algorithm is more efficient than bellman ford algorithm.

## V. SIMILARITY AND DISSIMILARITY OF DIJKSTRA ALGORITHM AND BELLMAN FORD ALGORITHM BASED ON GIS

Bellman ford and dijkstra both algorithms are used to find the shortest path of a network where the value of the shortest path may vary according to the situation of the network. We can solve cost and length problems using this both algorithm.

Dijkstra algorithm is faster than bellman ford algorithm. Bellman Ford algorithm is not suggested for larger networks. And in GIS, there are lots of data so we cannot suggest this algorithm. Bellman ford algorithm work in negative weighted graph and also detect negative cycle but in GIS distance is not in negative form. Second disadvantage is that Changes

in network topology are not reflected quickly but in GIS and in shortest path. we have to update quick fast the traffic information.so we cannot recommended for this algorithm. Bellman ford algorithm does not scale well. After this study we can say that dijkstra algorithm is very fast, and suitable for GIS application, so that dijkstra algorithm is widely used in real time application in GIS.

## VI. CONCLUSION

After studied these two algorithms I conclude that dijkstra algorithm is very faster than bellman ford algorithm. And also widely used in real time application in GIS. That's why I had chosen the Dijkstra's algorithm for finding shortest path in GIS technology. It is also work better and faster in very lager network

## REFERENCES

[1] DexiangXie, Haibo Zhu, Lin Yan, Si Yuan and JunqiaoZhang " An improved Dijkstra algorithm in GIS application" Proceedings of 2010 Conference on Dependable Computing, 2010

[2] David M. Mount "Design and Analysis of Computer Algorithms" Department of Computer Science, 2012

[3] GeeksforGeeks "Dynamic Programming" A computer science portal for geeks http://www.geeksforgeeks.org/dynamic-programming-set-23-bellman-ford-algorithm/

[4]DechuanKong ,Yunjuan Liang, Xiaoqin Ma, Lijun Zhang "Improvement and Realization of Dijkstra Algorithm in GIS of Depot

[5]Thippeswamy.K ,Hanumanthappa.J. , Dr.Manjaiah D.H. "A Study on Contrast and Comparison between Bellman-Ford algorithm and Dijkstra's algorithm".

[6] Wei Zhang ,Hao Chen , Chong Jiang , Lin Zhu,"Improvement And Experimental Evaluation Bellman-Ford Algorithm",International Conference on Advanced Information and Communication Technology for Education (ICAICTE 2013),2013