Implemented Modification in Dijkstra's Algorithm to Find the Shortest Path for 'N' Nodes with Constraint

Anjali Jain[1],U.Datta,Neelam.Joshi [1]

[1]Deptt. Of Computer Science & Engg. MPCT Gwalior.

Abstract

Roads play a Major role to the people live in various states, cities, town and villages, from each and every day they travel to work, to schools, to business meetings, and to transport their goods. Even in this modern era whole world used roads, remain one of the most useful mediums used most frequently for transportation and travel. The manipulation of a shortest paths between various locations appears to be a major problem in the road networks. The large range of applications and product was introduced to solve or overcome the difficulties by developing different shortest path algorithms. Even now the problem still exist to find the shortest path for road networks. To overcome the shortest path problem I make some changes and create new algorithm namely, Modified Dijkstra's Shortest Path using Priority Queue with Linked List algorithm using multiple parameters is proposed in this paper. The given algorithm is compared with the existing algorithm to prove its better efficiency.

I. INTRODUCTION

The Majorly use of Location Base Services and GPS technologies has used by researchers interest to develop a better and scalable shortest path algorithm to search a valid route for travelers and transporters over the road networks. Manipulating shortest path or distance between two or more points is play most fundamental and important role on road networks. Many researchers and people faces a lot of problem while planning their tour with their own vehicles. Recent days many applications and software were developed and introduce to solve the problem by finding an efficient route for the road networks. The past researchers shows their different shortest path algorithm to find the valid route for the road networks. But still the problem exist. Hence, the objective and goal of this research is to introduce a new shortest path algorithm to provide a best solution for the transporters and travelers over the road networks.

(GIS) geographic information systems is the most useful technology to determine the fastest route and dispatch an emergency vehicle like Military operations, fire, ambulance, service vehicle etc.

A lot of research has been done on improving the Dijkstra's algorithms, these research works are mainly pointed on searching the shortest path from one source to one destination in large scale road maps. The Navigation System which is the latest technology used this techniques to make it more reliable and accurate. Most of the research work on Dijkstra's Algorithm to make it a better and scalable shortest path algorithm. We work on large no of nodes to reduce the computational complexity and to provide the better results.

II.Related Work

Fuhao ZHANG [1] introduces the classical Dijkstra algorithm in detail, and describe the method of implementation of the algorithm and the disadvantages of the algorithm: it describes the adjacent node algorithm which is a better optimization algorithm based on Dijkstra algorithm. This algorithm makes correlation with each node in the various network topology information, and avoids the use of co-related matrix that contains huge infinite value, and making it more reliable and suitable analysis of the network for mass data. It is proved that this algorithm can save a lot of memory and is more suitable to the network with huge nodes,but in this research we found that as node grew larger this approach gets slow in searching nodes. Liu Xiao-Yan [2] this research used heap sort for unvisited nodes in geography network to improve the efficiency and reliability of Dijkstra algorithm but again its necessary to each and every time to arrange the heap(sorting) when node is inserted thus this research is also slow while inserting node in a heap. In Nikita Jaiswal's[3] paper they introduces the Dijkstra algorithm in detail, and illustrates the disadvantage of implementation of the algorithm ,They applied algorithm on Directed weighted graph to find shortest path between two nodes ,they worked on non-negative nodes, In this paper they also discuss about how they can improve Dijkstra algorithm in terms to determine path according to weight by increasing some no. of nodes.

[4] Mrs. Shweta Srivastava, Most of the algorithms modified to find out fruitful result using Dijkstra's algorithm. In this paper, they are going to do comparative analysis of some of the algorithms to solve this problem. They named Thorup's algorithm, adjacent node algorithm, a heuristic genetic algorithm, augmented shortest path, and improved better version of the Dijkstra's algorithm and a graph partitioning based algorithm. But this algorithm was very complex to sort out the Dijkstra's problem.[5] N. Ravi Shankar N. In this research they used Critical Path Method (CPM) is to find out critical activities on the critical path so that resources may be use less time to find out the result. To find out the critical path, three parameters such as latest event time , earliest event time, and slack time for each of its activities are find out. They modified Dijkstra's algorithm for critical path method to find latest event time, slack time, earliest event time for each of its activities in a project network. [6] Charika Jain.,In this paper they read out how to select a path with the minimum cost in terms of expected end-to-end delay in a network.They worked on the transmission delay and queuing delay in buffer. [7] In this paper, they formulate the quantum algorithm for the Dijkastra's shortest path algorithm and propose as Quantum Dijkastra algorithm (QDA) and propose its quantum circuit design, which is first of its kind. Implementing QDA the find out good result but again the major problem is that whenever we insert new node it must be optimized using QDA and they apply Dijkastra's shortest path.

Graph Theory

A graph G is a collection of two sets V & E where V is the collection of vertices $v_{0,}$ $v_{1,}$..............v_{n-1} also called nodes and E is the collection of edges $e_{1,}$ $e_{2,}$.............e_n where an *edge is an area which connects two nodes. This can be represented as –*

G=(V,E)

V(G)=(v$_{0,}$v$_{1,}$.........v$_{n}$)or set of vertices

E(G)=(e$_{1,}$e$_{2,}$..........e$_{n}$)or set of edges

Overview of Shortest Path Algorithm

There are several cases in graph where we have a need to know the shortest path from one node to another node. General electric supply system and water distribution system also follow this approach. The best example we can take is of a railway track system. Suppose one person wants to go from one station to another then he needs to know the shortest path from one station to another here station to another. Here station represents the node and tracks represent edges. In computers. It is very useful in network for routing concepts.

There can be several paths for going from one to another node. But the shortest path is that path in which the sum of weights o the included edges is the minimum. There are several algorithms to find out the shortest path. All-Pairs shortest path algorithms and it is shown in below figure 1.

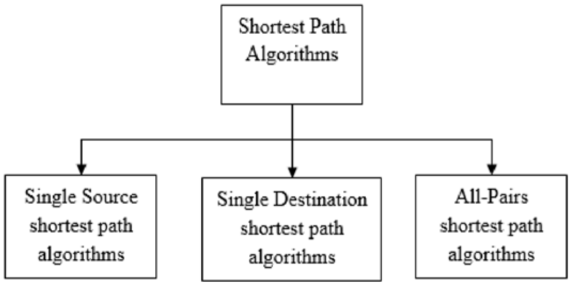
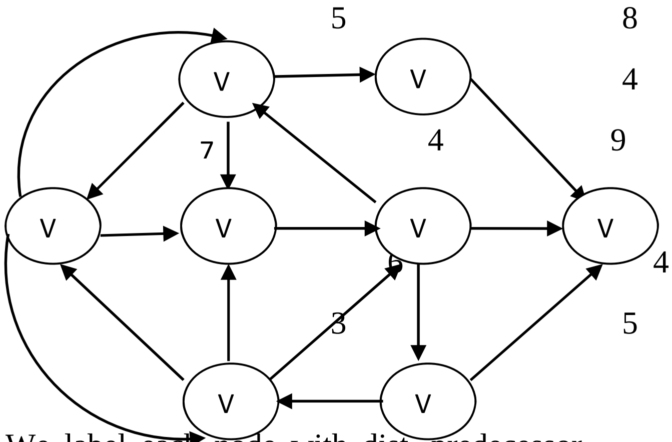

Shortest Path Algorithms
Single Source shortest path algorithms
Single Destination shortest path algorithms
All-Pairs shortest path algorithms

Figure1. Classification of Shortest Path Algorithm

Djikstra's Shortest Path Algorithm

Here we will describe the Dijkstra's algorithm Let us take a graph and find out the shortest path from the source node to destination node.



We label each node with dist, predecessor and status. Dist of node represents the shortest distance of that node from the source node, and predecessor of node represents the node which precedes the gives node in the shortest path from source. Status of a node can be permanent or temporary. In the figure, shaded circles represent permanent nodes. Making a node permanent means that it has been included in the shorts path. Temporary nodes can be relabeled if required but once a node is made permanent. it can't be relabeled.

Time Complexity

The execution time of begin for loop is *O(V)*. In each iteration of the while looping, Find out Min of the heap is *logV*. The inner most for loop iterates for each adjacent node of the current node, the total execution time is *O (E)*. And therefore, the time complexity of given algorithm is *O((V + E)*log(V) = O(E* log(V))*. The accuracy of this algorithm is well proved in [5]. When the number of nodes in a graph increases, the execution time of the given algorithm will become longer and longer. Usually, a road network of a town or city has more than 10^4 nodes. A fastest shortest path algorithm becomes find out to analyze the existing different Dijkstra's shortest path algorithm. We found that in the existing Dijkstra's algorithm.

III. PROPOSED ALGORITHM

Manipulating shortest path or distance between source and destination is one of the most fundamental and important key problems on road networks. Roads play a Major role to the people live in various states, cities, town and villages, from each and every day they travel to work, to schools, to business meetings, and to transport their goods. Even in this modern era whole world used roads, remain one of the most useful mediums used most frequently for transportation and traveling. The exact manipulation of a shortest paths between various locations appears to be a major problem in the road networks. The large range of applications and product was introduced to solve or overcome the difficulties by developing different shortest path algorithms.Many Researchers, work numerously on shortest path algorithms like A* search, Manipulating shortest path or distance between source and destination is one of the most fundamental and important key problems on road networks. Roads play a Major role to the people live in various states, cities, town and villages, from each and every day they travel to work, to schools, to business meetings, and to transport their goods. Even in this modern era whole world used roads, remain one of the most useful mediums used most frequently for transportation and travel. The manipulation of a shortest paths between various locations appears to be a major problem in the road networks. The large range of applications and product was introduced to solve or overcome the difficulties by developing different shortest path algorithms.

Many Researchers, work numerously on shortest path algorithms like A* search, Dijkstra's , Bellman-Ford and Johnson's algorithm were developed. Finally, researchers was found that Dijkstra's shortest path algorithm is the best algorithm for calculating shortest path in road or transportation networks. In existing Dijkstra's algorithm have some modification to improve the efficiency and accuracy to find a better shortest path and to reduce the time complexity. Hence, a new algorithm called Modification In Dijkstra's Algorithm To Find The Shortest Path For 'N' Nodes With Constraint (MDSC) is proposed Comparing QDA with MDSC

In this paper, we propose some amendment on QDA algorithm in order to optimize it by limit the number of iterations. The main logic is to solve the problem where more than one node satisfies the condition of the next step in the existing Dijkstra's algorithm. After applying the proposed modifications, the maximum number of loops or iterations in Dijkstra's algorithm is lesser than the number of the nodes.

Dijkstra's algorithm, Bellman-Ford algorithm algorithm and Johnson's algorithm were developed. Finally, researchers was found that Dijkstra's algorithm is the best algorithm for manipulating shortest path in road networks. But the Dijkstra's algorithm have some modification to improve the efficiency and accuracy to find a better shortest path and to reduce the time complexity. Hence, a new algorithm called Modification In Dijkstra's Algorithm To Find The Shortest Path For 'N' Nodes With Constraint (MDSC) is proposed Comparing QDA with MDSC.

Quantum computers use quantum mechanical properties to speed up the process and decreases computation time and result oriented processing capabilities compared to basic computing. If this algo is implemented to find the shortest path on map its gives us fruitful result. But I have found that the Quantum Dijkastra algorithm (QDA) is reliable then Dijkstra Algo but it used some extra iteration to find out the result.

In this paper, we propose some amendment on QDA Algo in order to optimize it by lesser the number of iterations. The main logic is to solve the problem where more than one node satisfies the criteria of the second step in the existing Dijkstra's algorithm. After applying the proposed modifications, the maximum number of loops and iterations of Dijkstra's algorithm is less than the number of the graph's nodes.

We will present a modified version of Dijkstra's Algorithm for n vertices and n *adjacency matrix* in order to lesser the total number of iteration by optimized the situation of many shortest paths:

1.Each vertex v that is not yet permanently labeled gets a new temporary label whose value is given by min[old label of v, (old label of c + dcv)], in n adjacency matrix where c is the current vertex permanently labeled, in the previous iteration, and dcv is the given distance between two vertices c and v. If c and v are not joined by an edge, then dcv=infinity .

2.The smallest value among all the temporary labels c is found and store them into priority queue with link list to get the shortest path, and this becomes the permanent label of the corresponding vertex. In case of more than one shortest path, select all of them for permanent labeling.

3.Steps (1) and (2) are repeated alternately maximum n-1 times until the destination vertex t gets a permanent label.

Implementation in Java

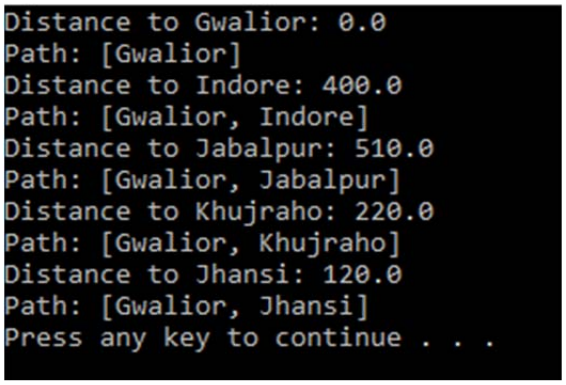
[Member function which compute and compare path]

```
 public static void computePaths(Vertex source)
  {
        source.minDistance = 0.;
        PriorityQueue<Vertex> vertexQueue = new PriorityQueue<Vertex>();
        vertexQueue.add(source);

        while (!vertexQueue.isEmpty())
        {
        Vertex u = vertexQueue.poll();

            // Visit each edge exiting u
            for (Edge e : u.adjacencies)
            {
                Vertex v = e.target;
                double weight = e.weight;
                double distanceThroughU = u.minDistance + weight;
                if (distanceThroughU < v.minDistance)
                {
                    vertexQueue.remove(v);
                v.minDistance = distanceThroughU ;
                v.previous = u;
                vertexQueue.add(v);
            }
        }
    }
}
```

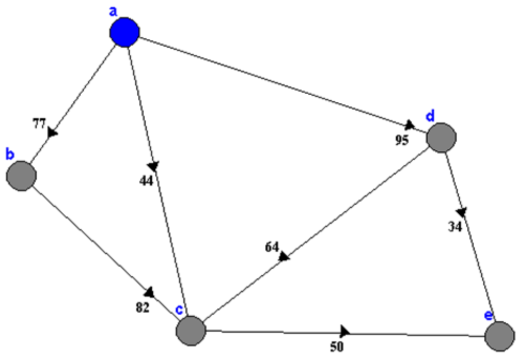
[Function to get the shortest path]

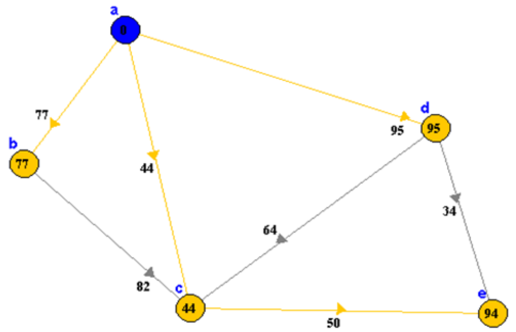
```
public static List<Vertex> getShortestPathTo(Vertex target)
{
    List<Vertex> path = new ArrayList<Vertex>();
    for (Vertex vertex = target; vertex != null; vertex = vertex.previous)
        path.add(vertex);
    Collections.reverse(path);
    return path;
}
```

Result



```
Distance to Gwalior: 0.0
Path: [Gwalior]
Distance to Indore: 400.0
Path: [Gwalior, Indore]
Distance to Jabalpur: 510.0
Path: [Gwalior, Jabalpur]
Distance to Khujraho: 220.0
Path: [Gwalior, Khujraho]
Distance to Jhansi: 120.0
Path: [Gwalior, Jhansi]
Press any key to continue . . .
```





IV. CONCLUSION

We can lesser our cost when we build a graph. It is because the Dijkstra's will find the shortest path weight from one source node to other node. Therefore, we need not build much of router to build path from a node to other. This algorithm also can increase the performance to find out the shortest route with minimum iterations. The algorithm will find the minimum cost. Path weight is propagation delays for a system, Best Suited for Robot path planning, Logistics Distribution Lines, Link-state routing protocols.

References

1]. Fuhao ZHANG*, Ageng QIU, Qingyuan LI ,IMPROVE ON DIJKSTRA SHORTEST PATH ALGORITHM FOR HUGE DATA Chinese Academy of Surveying and Mapping, Beijing, China, 100039, Zhangfh@casm.ac.cn

2]. Liu Xiao-Yan, Chen Yan-Li, Application of Dijkstra Algorithm in Logistics Distribution Lines School of computer science and technology,Henan Polytechnic University,JiaoZuo, China

3]. Nikita Jaiswal , Rajesh Kumar Chakrawarti ,Increasing no. of nodes for Dijkstra algorithm without degrading the performance,COMPUTER SCIENCE AND ENGGINEARING Shri vaishnav institute of technology and science Indore

4]. Mrs. Shweta Srivastava
Comparative Analysis of Algorithms for Single Source Shortest Path Problem
shwetasrivastava21@gmail.com Computer Science & Engineering Department, ABES Engineering College Ghaziabad. India

5]. N. Ravi Shankar and V. Sireesha
Using Modified Dijkstra's Algorithm for Critical Path Method in a Project Network
Department of Applied Mathematics, GIS, GITAM University,Visakhapatnam, India

6]. Charika Jain* , Jitendra Kumawat
International Journal of Advanced Research in Computer Science and Software Engineering Research Paper
Available online at: www.ijarcsse.com
Processing Delay Consideration in Dijkstra's Algorithm Computer Science, Amity University Rajasthan Rajasthan, India

7]. Paramita Ray
Quantum Simulation of Dijkstra's Algorithm
A.K.C School of Information Technology Calcutta University Calcutta – India
International Journal of Advance Research in Computer Science and Management Studies
Volume 2, Issue 9, September 2014