Modification In Dijkstra's Algorithm To Find The Shortest Path For 'N' Nodes With Constraint

Anjali Jain[1],U.Datta[1],K.K.Joshi[1]
[1] Deptt. of Computer Science &Engg., MPCT Gwalior.

Abstract

Modification In Dijkstra's Algorithm To Find The Shortest Path For 'N' Nodes With Constraint" is software which is used to find out the shortest distance between from source to your destination.If you want to know the distance from source to your destination with constraints anywhere in India and don't know where to look for it, here is the solution to solve your problem. A search engine which let you find the distance between Indian cities with constraints within seconds. Use of this software is to find locations, calculate the shortest path to your destination node as well as to find out your positional co-ordinates. Select the cities as per your requirement, the algorithm would find out the overall distance between the source and distance with constraints.

Keyword:,Graph,Distance Vector , Adjacency list, Adjacency Matrix.

1. Introduction

In graph theory, the **shortest path problem** is used to find the path between source vertices and destination vertices (or nodes) in a **graph** such that the sum of the weights of its constituent edges is minimized.

What is *Graph*?

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 arc which connects two nodes. This can be represented as :

G=(V,E)

V(G) = (v_0,v_1,………v_n) or set of vertices

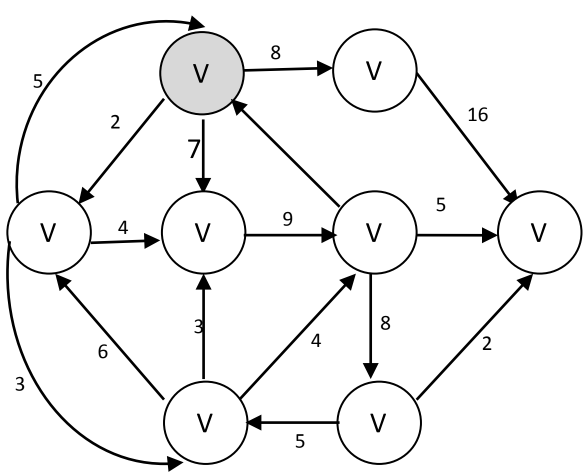
E(G) = (e_1,e_2,……….e_n) or set of edges

We have mainly two components in graph, nodes & edges. now two have to design the data structure to keep these components in mind. There are two ways for representing the graph in computer memory. First one is the sequential representation and second one is linked list representation. Adjacency matrix is the matrix, which keeps the information of adjacent nodes. In other words, we can't say that this matrix keeps the information that whether this node is adjacent to any other node or not. If the adjacency matrix of the graph is sparse then it is better to represent the graph through adjacency link list.

In adjacency link list representation of graph, we will, maintain two lists. First list will keep information of all the nodes in the graph and second list will maintain a list of adjacency nodes for each node.

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 determine or find the shortest path. 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.

The procedure is as -
1. Initially make source node permanent and make it the current working node. All other nodes are made temporary.
2. Examine all the temporary neighbors of the current working node and after checking the condition for minimum weight, reliable the required nodes.

3. From all the temporary nodes, find out the node which has minimum value of dist , make this node permanent and now this is our current working node.

4. Repeat steps 2 and 3 until destination node is made permanent.

Dijkstra's Algorithm Advantages

1) We can minimize our cost when we build a network. It is because the Dijkstra's will find the shortest path value from a given source node subject to destination node. Therefore, we need not build much of router to build path from a node to other.

2) Dijkstra's algorithm also can maximize the efficiency of your system. The algorithm will find out the minimum path value. Path weight is propagation delays for a system.

Best Suited For

1) Robot path planning

2) Logistics Distribution Lines

3) Link-state routing protocols

An example is shown here for actual working of Dijkstra's algorithm:

2. Proposed work

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. Here we will describe the Dijkstra's algorithm.

I am implementing totally different concept to find out shortest path using Scaled map of India. The map which we are using is scaled map which return correct distance between two cities with as many constraints. Here we will use link list to store and traverse N nodes or vertices. We label each node (city) 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 given node in the shortest path from source. Status of a node can be permanent or temporary.

The procedure is as -

1. Initially make source node permanent and make it the current working node. All other nodes are made temporary and store it into priority queue using link list and set constraints with each node which is not set within the graph.

2. Examine all the temporary neighbors of the current working node and after checking the condition for minimum weight, reliable the required nodes.

3. From all the temporary nodes, find out the node which has minimum value of dist , make this node permanent and now this is our current working node.

4. Repeat steps 2 and 3 until destination node is made permanent

3.Research Design problem and Formulation

RESARCH DESIGN PROBLEM

Earlier researchers have performed various operations on Dijkstra's algorithm to determine the shortest path between the nodes and have got the better results in their researches for the specified number of nodes.

But, the results were limited to the number of nodes fixed at the time while declaring the size of the data structure.

Fuhao ZHANG [1] introduces the classical Dijkstra algorithm in detail, and describe the useful process of implementation of the algorithm and drawbacks 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 different network topology and 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 space and is more reliable 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 introduce as Quantum Dijkastra algorithm (QDA) which gives fruitful result in quantum network and 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.

RESEARCH FORMULATION

To resolve this problem and to give the best solution for increasing number of nodes we will be replacing the data structure with the priority queue link list having the capacity to store N nodes with some constraints containing the value of predecessor , status and distance.

5. Conclusion

We have discovered an improved version of the algorithm that makes only two calls to the replacement paths subroutine after each new path is discovered. Currently, our algorithm used priority queue with link list with some constraints, plus one Dijkstra call. This change should improve the running time of our algorithm by about 40%.

References

1]. Fuhao ZHANG*, Ageng QIU, Qingyuan LI

Chinese Academy of Surveying and Mapping,
IMPROVE ON DIJKSTRA SHORTEST PATH ALGORITHM FOR HUGE DATA
Beijing, China, 100039, Zhangfh@casm.ac.cn

2]. Lines Liu Xiao-Yan, Chen Yan-Li
Application of Dijkstra Algorithm in Logistics Distribution 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 Engineering Department, 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