International Journal of Mathematics and Physical Sciences Research ISSN 2348-5736 (Online)
Vol. 4, Issue 2, pp: (141-145), Month: October 2016 - March 2017, Available at: www.researchpublish.com

Shortest Path Algorithms

[1]S.P. Behera, [2]S. Bhattacharjee, [3]D. Mishra

Department of Mathematics C.V Raman (Autonomous) Engineering College, Bhubaneswar, Odisha
Ravenshaw University, Cuttack, Odisha

Abstract: **The shortest path and minimum spanning tree problems are two of the classic problems in combinatorial optimization. Using Dijkstra algorithm for searching shortest path problem is an important content of the application of GIS (Geographic Information System), but it fails to find all the shortest paths. In this work the article based on the Dijkstra algorithm, added some data structure and proposed an algorithm that calculates all the shortest path of one vertex to others, the data structure is relatively simple; while effectiveness of the algorithm is explain through a numerical example.**

Keywords: **Graph, Weighted graph,adjacent matrix, Disjkstra algorithm.**

1. INTRODUCTION

Dijkstra algorithm was proposed by Dick Stella, a computer scientist from the Netherlands, in 1959. From one vertex to the rest of the vertices of the shortest path algorithm, the solution is to map the shortest path problem. For a directed graph or undirected graph G, each edge add a real number ω (e), called ω (e) right on the edge e, G, together with an additional edge in the real number is called a weighted graph. Usually expressed as a weighted graph: G = (V, E, W), where, V is the set of vertices, E is the set of edges, W is the set of weight of corresponding edge. That is the meaning of weight: Such as highway mileage between cities, or the costs of building roads, or travel needed to fuel consumption. Obviously, in a weighted graph G = (V, E, W), in the weight of each edge ei, W(ei) must be greater than zero. All channels from vertex u to v, the smallest sum of the weight path is the shortest path from u to v, find a given shortest path between two points is called the shortest path problem [9].

2. PRELIMINARIES

In this section we provide some basic definition & Concepts

***Graph*:** A graph G consists of a set V of vertices (nodes) and a set E of edges (arcs).We write G = (V, E), V is a finite and non-empty set of vertices. E is a set of pairs of vertices, their pairs are called edges.

***Weighted graph*:** A graph G is said to be weighted if each edge e in G is assigned a non-negative numerical value w (e) called the weight or length of e.

***Adjacency matrix*: It** is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph

3. IMPROVED ALGORITHM FOR SHORTEST PATH

3.1 Dijkstra Algorithm Theory:

Dijkstra algorithm was proposed by Dick Stella, a computer scientist from the Netherlands, in 1959. From one vertex to the rest of the vertices of the shortest path algorithm, the solution is to map the shortest path problem. For a directed graph or undirected graph G, each edge add a real number ω (e), called ω (e) right on the edge e, G, together with an additional edge in the real number is called a weighted graph. Usually expressed as a weighted graph: G = (V, E, W), where, V is the set of vertices, E is the set of edges, W is the set of weight of corresponding edge. That is the meaning of weight: Such as highway mileage between cities, or the costs of building roads, or travel needed to fuel consumption. Obviously, in a weighted graph G = (V, E, W), in the weight of each edge ei, W(ei) must be greater than zero. All channels from vertex u to v, the smallest sum of the weight path is the shortest path from u to v, find a given shortest path between two points is called the shortest path problem [9].

3.1.1 Mathematical Description of the Dijkstra Algorithm [9]:

Let $G=(V,E,W)$ is a non-negative weight network, $V=(v_1,v_2,\ldots,v_n)$. The length of the shortest path in $G(v_i,v_j)\in E$ satisfy the equation:

$$u_1=0$$
$$u_j=\min(u_k+w_{kj}) \qquad (j=2,3,\ldots,n) \qquad (1)$$

If G, from the vertex v_1 to the rest of the vertices, the shortest path length sorting by size:

$$u_{i1}\le u_{i2}\le\ldots\le u_{in}$$

Here, $i_1=1, u_{i1}=0,$ Then from (1) have:

$$u_{ij}=\min_{k\ne j}\{u_{ik}+w_{ikij}\}$$

$$=\min\{\min_{k<j}\{u_{ik}+w_{ikij}\},\min_{k>j}\{u_{ik}+w_{ikij}\}\} \qquad (j=2,3,\ldots,n)$$

When $k>j, i\ne j, u_{ik}\ge u_{ij,}$ and $w_{ikij}\ge 0,$ thus

$$u_{ij}\le u_{ik}+w_{ikij}$$

That is $u_{ij}\le\min_{k>j}\{u_{ik}+w_{ikij}\}$

Therefore $u_{ij}=\min_{k<j}\{u_{ik}+w_{ikij}\} \qquad (j=1,2,3,\ldots,n)$

u_{ij}, one of the solution $(u_{i1},u_{i2},\ldots,u_{in})$ is the shortest path length of $G(u_i,u_j)$.

3.1.2 Calculation Steps and the Problems of Dijkstra Algorithm:

Weighted graph can be expressed as adjacency matrix cost[i] [j], which states: if between v_i and v_j have no direct path, the cost[i] [j] = ∞; if Between v_i and v_j have direct path, the cost[i] [j] = w_{ij}; if i = j, then the cost[i] [j] = 0. The set of S storage initial source of shortest path, during the process of calculating, the vertex have determined the shortest path added to the S. Final dist[i] storage the shortest path length, the source point to the vertex, specific steps are as follows:

(1) Initialize S and dist.

S= $\{v_0\}$, dist[i]=cost[0][i], i=0, 1, …, n- 1.

(2) Select v_j, so dist[j]=min{dist[i]| $v_i\in(v-s)$ };

$$s=s\bigcup v_j$$

(3) Modify the length of the shortest path from v_0 to $v_k\in(V-S)$.

If dist[j] + cost[j][k]<dist[k],

So dist[k]= dist[j] + cost[j][k].

(4) Repeat steps (2), (3), until obtain the shortest path that from the source point v_0 to the rest of vertices v_j .

However, Dijkstra algorithm can only find one shortest path from v_0 to v_j , and cannot calculate all the shortest paths. In this thesis, a slightly improved of Dijkstra algorithm is provided in more intuitive way to solve a vertex to other vertices of all shortest paths [10].

3.2 Improved Algorithm and Examples:

(1) The introduction of Dijkstra algorithm, that is, step (1) to step (4) in the 2.3;

(2) According to the cost and the dist, creating a correction matrix corr.

Method: If 0< cost[i][j]< ∞ ,

then corr[i][j]= cost[i][j] + dist[i];

otherwise， corr[i][j] = cost[i][j];

(3) Create a successor node set succ(v_i).

Method: By succ(i) ={J|corr[i][j] = dist[j] and i $\neq$ j}

Find the set that the successor node for each vertex ;

(4) According to the corr and succ output the shortest path from the source to all other vertices.

The algorithm can find from one vertex to all other vertices the shortest path. The following detailed description of the algorithm implementation through the example of digraph and undirected graph.

3.2.1 Example:

Figure.2 shows a directed graph G1, to find the shortest path from vertex v_0 to all other vertices the specific steps are as follows:

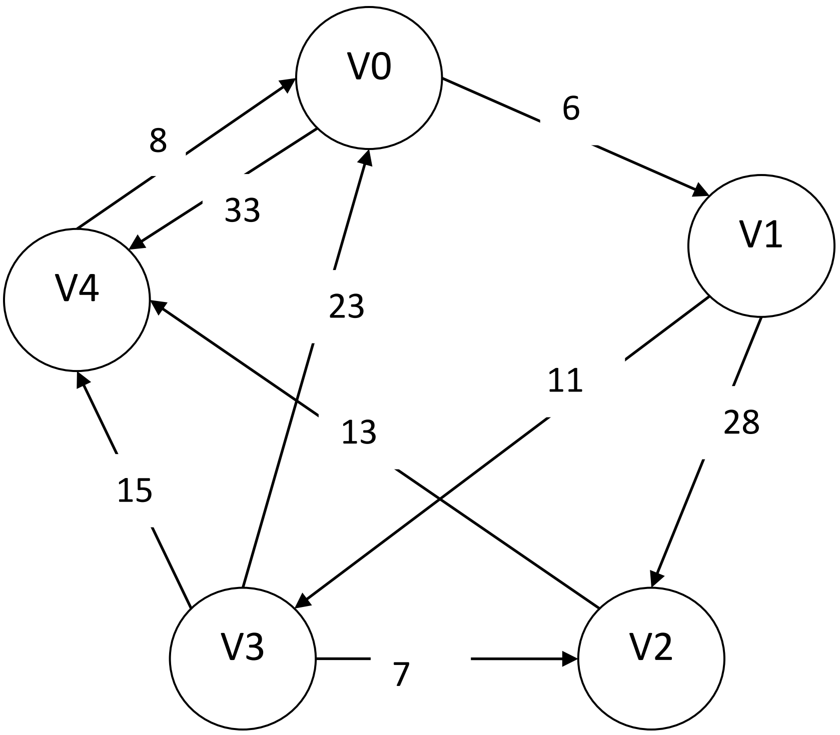

Fig. 2 Weighted directed graph G1

(1) Adjacency matrix is given:

$$\begin{bmatrix} 0 & 6 & \infty & \infty & 33 \\ \infty & 0 & 28 & 11 & \infty \\ \infty & \infty & 0 & \infty & 13 \\ 23 & \infty & 7 & 0 & 15 \\ 8 & \infty & \infty & \infty & 0 \end{bmatrix}$$

Initialize S and dist [i] ($0 \le i \le 4$) : S= {v_0}, dist [i]= { 0, 6,∞,∞, 33};

(2) based on dist [i] = min{ dist [i] | $v_i \in V - S$ }, to find dist[1] = 6; S={ v_0 , v_1};

(3) modify the shortest path length, starting from v_0 to any node on the V-S v_k (then $2 \le k \le 4$) , dist[i] = { 0, 6, 34,17, 33};

(4) dist[3] = min{ dist[i] |vi∈V- S} = 17; S={ v_0 , v_1 , v_3 } ; dist [i] = { 0, 6, 24, 17, 32}.

(5) dist[2] = min{ dist[i] | v_i∈V- S} = 24; S={ v_0 , v_1 , v_2 , v_3} ; dist[i] = { 0, 6, 24, 17, 32}.

Although at this point there is a vertex not incorporated in set S, but it is the shortest distance have been determined, so the whole operation is over.

(6) Created corr matrix through the cost and dist :

$$\begin{bmatrix} 0 & 6 & \infty & \infty & 33 \\ \infty & 0 & 34 & 17 & \infty \\ \infty & \infty & 0 & \infty & 37 \\ 40 & \infty & 24 & 0 & 32 \\ 40 & \infty & \infty & \infty & 0 \end{bmatrix}$$

According to corr, dist and succ(i) = { j | corr[i] [j] = dist[j] and i≠j}, find the collection from the successor node for each vertex, here: succ(v_0) = { v_1 } , succ(v_1) = { v_3 }, succ(v_2)= {null}, succ(v_3) = { v_2 , v_4 }, succ(v_4) = {null}. Then the vertices v_0 to all other shortest path is $v_0v_1; v_0v_1v_3v_2; v_0v_1v_3; v_0v_1v_3v_4$.

4. CONCLUSION AND FUTURE SCOPE

4.1 Conclusion:

We have presented an improved Dijkstra algorithm that calculates all the shortest path of one vertex to other vertex.

4.2 Future Scope:

We also improve shortest path algorithm to compute the shortest path based on network matrix of Dijkstra. Many $N \times N$ arrays are defined to store graphical data and compute. N is referred to the number of the network nodes. When the number of the nodes is very large, it occupies a lot of CPU memory. For example, when the number of the nodes is 3000, it needs $4 \times 3000 \times 3000$ =36000000 bytes=36 MB memory and if the number of nodes is 6000, it needs 144 MB memory. Therefore, we try to improve Dijkstra algorithm for the network model having huge data.

REFERENCES

[1] Kowalik L. and Kurowski M., Short path queries in planar graphs in constant time. In Proc. 35th ACM Symp. on the Theory of Computing, 2003.

[2] Meyer U. and Sanders P., Parallel shortest path for arbitrary graphs. In Proc. 6th Int'l Euro-Par Conference, LNCS 1900, pages 461-470, 2000.

[3] Meyer U. and Zeh N., I/O-efficient undirected shortest paths. In Proc. 11th Ann. Euro. Symposium on Algorithms (ESA), 2003.

[4] Roberto Cordone, Gianluca Passeri "Solving the Quadratic Minimum Spanning Tree Problem *Applied Mathematics and Computation*, *Volume 218, Issue 23*, *1 August 2012*, *Pages 11597-11612*.

[5] Francis J. Vasko, Robert S. Barbieri, Brian Q. Rieksts, Kenneth L. Reitmeyer, Kenneth L. Stott Jr. "The cable trench problem: combining the shortest path and minimum spanning tree problems "*Computers & Operations Research*, *Volume 29, Issue 5*, *April 2002*, *Pages 441-458*

[6] Wang, K.-Y., Zhao, C.-J., Xu, G.-X., et al.: A High-efficiency Realization Way of the Shortest Path Search Problem in GIS Field. Journal of Image and Graphics 8(8), 951–956 (2003)

[7] Cheng, S., Li, Y., Chen, X.-G.: Research and Application of Dijkstra Algorithm in GIS. Journal of Anyang Institute of Technology 9(2), 43–45 (2010)

[8] Tang, J.-W.: Improving Dijkstra Algorithm about the Shortest Path in GIS Field. Journal of Weinan Teachers College 21(2), 51–54 (2006)

[9] Zhang, F.-H., Liu, J.-P., Li, Q.-Y.: A New Way of Network Analysis Based on Dijkstra. Technology of GIS 2, 38–41 (2004)

[10] Liu, P., Bai, C.-M.: An improved algorithm of shortest paths based on dijkstra algorithm. Journal of Qinghai Normal University (Natural Science) 1, 79–81 (2008)