Fast Algorithms To Find The Shortest Path Using Matrices

Wafaa Mustafa Hameed, Asan Baker Kanbar, Jumah Aswad Zarnan

Abstract: Shortest path algorithms are among the most studied network flow optimization problems introduced to in such approach such as ants colony, Greedy, Floyd, K and k* algorithms. In this paper we introduce a new algorithm to find the shortest path between two centers (nods), these nods may be represent a fire zone, bus station, cities…etc.In this algorithm nods will be represents as a matrix, the number of columns and Rows equal to the number of arcs of the network. The elements represent as 0 that means that there's no arc between the centers (nods), or 1 that's mean there's an arc between the two centers. The solving of this matrix occurs in many stages: Stage1: specifying the number of arcs in the network by calculating the number of elements equal to 1 as arrow of matrix. Stage2: specify the number of arcs by identifying the number of rows and columns hold the value 1. Stage3: count the length of the paths depending on the arcs which is installation the paths and compare between these paths to specify the shortest path. So in this algorithm we need just to inter the number of arcs, start state node and goal state node of the network, we choose VC++ as a language to program this application in, a compiled program will always be faster than an interpreted program.

Index Terms: shortest path, matrix, nods, arcs, network, algorithms, optimization

---

1 Introduction

Shortest path finding problems are most encountered problem in graph algorithms and communication network applications. Since finding shortest path over network topology is demanding and expensive, it is worthy to consider various technique and heuristics that can help in improving existing algorithms [1]. Many papers study algorithms to find the shortest path problem [2][3][4][5][6][7][8][9][10][11]. The purpose of this paper is to introduce new algorithms to find the shortest path, the importance of this algorithm that it can find the shortest path irrespective of the size of the network and its generis. This algorithm achieved several aims (1) make the search about the shortest path using a software program (2) manipulate the dual path networks (3) manipulate the network howsoever the number of arcs between the centers. Using the program which is built using VC++ programming language we just need to input the number of arcs of the network, the number of the start state of the arc and the number of its end node and the program will make a processing for the matrix which it is columns and rows spring up from the path of the network, the number of columns and rows equal to the number of the arcs in network, the elements of the matrix consist from 0's which it is mean that there's no arc's between the nodes and 1's which it is mean that there's an arc between the nodes. The program will specify the tracks of the network whatever it is nested and determine the arcs which it is a part of each path and the length of the path. Then select the shortest path of the network.

- *Wafaa Mustafa Hameed, Asan Baker Kanbar, Jumah Aswad Zarnan*
- *Asst. Lect., Dept. Of Computer Science, Cihan University, Sulaimaniya, Kurdistan Iraq* afafa_ju223@yahoo.com
- *Asst. Lect , Dept. Of Computer Science, Cihan University, Sulaimaniya, Kurdistan Iraq* esenturkay@hotmail.com
- *Asst. Prof. Dr., Dept. Of Accounting by IT, Cihan University, Sulaimaniya, Kurdistan Iraq* jumah_223@yahoo.com

2 The Definition Of The Algorithm Referred In This Research

In this algorithm we represent the matrix as:

Stage 1:-

1) Determine the number of arcs, we denoted to it as n.
2) The number of arcs starts from 1 to n.
3) The arc I connect between two nods, arc(I,1) for the beginning node of the arc and the arc(I,2) for the end node of arc. N=1, 2, .., n.
4) The value of arc A(i) (depend on the problem under study may be consider cost, time or distance).

In the other step we add arc(0,1) and the end node arc(0,2) =1 and also the beginning node arc(1,1) =1.
The arc (n+1) added to the end (goal) node.
Arc (n+1, 1) = arc (n, 2) =k
The value of the end node K+1
So in this case (n+1, 2) =K+1

Stage2:-

So the matrix (n+1)*(n+1) founded and as we mentioned before the number of the rows from 0 to n and columns take the values from 1 to (n+1) then X(i, j)=1
If we have an arc j locate after arc I then it will take the value 1 and the elements take the value X(I, j)=0, otherwise i=0,1,….,n and j=1,2,…..,n+1.

Stage3:-

To specify the number of arcs let us consider the path is the number of the paths and consider L(i) to the number of rows I which has the value 1.
Consider track=0
Start from i=0, then L(0)=0; count the number of elements which its values=1, so the number of track and L(0) equal to the number of elements which it is =1.
If X(I, j)=1; j=i+1,i+2,…..,n+1 then L(i)=L(i)+1, path =path+1 in the first row.
In the second row I=I+1 consider l(i)=0; count the number of elements which they hold the value 1, then the value of L(i) became equal to the number of elements hold the value 1,
if (X(I, j))=1, j=i+1,i+2, …., n+1 then L(i)=L(i)+1
if L(i)>1 then the number of paths increased to become in value equal to the number of elements which is equal 1
if X(i, j)=L ; j=i+1,i+2,…….,n+1 then L=L(i)+1.

Now if L(i)>1 then the number of paths will increased in value equal to the number of elements which it is equal to 1 minus 1
If L(i)>1 then the path = path+(L(i)-1) which it mean that there is more than one arc go from the node arc(I, 2)
Repeating the two steps above till i=n then at that's mean we specifies the number of arcs after each arc I through L(i) and we able now to count the number of paths in network through the value of path.

Stage4:

Specify the number of paths and also the arcs for each path.
Consider:

S= the number of links in path.

B=path's number

C=column's number

$R1$=row's number

$C2$= column from previous stage.

$V(b,s)$ number of links in path.

So the number of arcs in each track can be found as follows.

1. Let b=1 , S=0
2. From the first row i=0, consider x(0,j) which it is the first element=1; j=1, 2, 3, …,n+1, with keeping the number of the column and row for the first element hold the value1.
   R1=I, C2=j , C=j; change the value of the element to zero X(0,j)=0.

Note:- if all the value of elements in the first row=0 then stop tracking, and that's mean we found all the paths in network.

3. If i=c that's mean

$L(i) = 1$, that's mean all the elements of the first row =1 the

- Specify the column of elements 1 $C = j$
- Let S=S+1, V(b, s) that's mean the arc i consider in the path.
- If C=n+1 then apply the step 3 or go to step 4.

A. If we have more than one arc go from the same node, $L(i) > 1$; in this case
   a. Specify the column which is contain the first element =1 in the row $C = j$.
   b. Consider S=S+1, $V(b,S)$, that's mean the arc I will be consider as an elements in path.
   c. Let $X(I,j) = 0$, that's mean the value of the selected elements =0.
   d. Let $L(i) = L(i) - 1$, that's mean decreasing the number of elements in that row in value equal to (-1).
   e. Consider the elements $X(R1,C2)$ in the previous stage and let it =1, $X(R1,C2) = 1$.
   f. Let R1=L(R1)+1 that led to increasing of elements in the row with value 1.
   g. Keeping the number of column on the row of the elements which it is equal 0 in step c . R1=I , C2=j. if C=n+1 then go to step d, else go to step c.

4. Then
   a. Let the path $(b) = S$ booking the number of arcs in path b.
   b. Let b=b+1 specifying the arcs of the next path.
   c. Let S=0, then back to step b.

Stage5:-

To determine the shortest path:

1. Determine the length of all paths from the following equation:

$$all - paths(b) = \sum_{s=1}^{path(b)} AC(i) * \left(V(b,S)\right) \quad , \quad b = 1, 2, \dots\dots, path$$

2. Determine the shortest-path=Min [all-path(b)] , b=1, 2, …….., path

3. Experimental Part:

Let us consider the following network in figure (1), we will use this network and apply the proposed algorithm on it

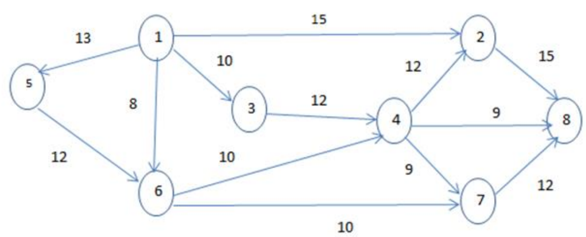


Fig1: *the sample network*

First: the first stage is to inter the number of arcs, and the number of the nodes and the end node in the path as in table 1

TABLE 1

Inter The Number Of Arcs, Nods And End Node

arcs i	Stand node of the path arc(I,1)	The end node of the path arc(I,2)	The value of arc A(i)
1	1	2	15
2	1	3	10
3	1	5	13
4	1	6	8
5	2	8	15
6	3	4	12
7	4	2	12
8	4	7	9
9	4	8	9
10	5	6	12
11	6	4	10
12	6	7	10
13	7	8	12

***Second**: depending on the first stage the program will form the following matrix as in Figures 2, 3*

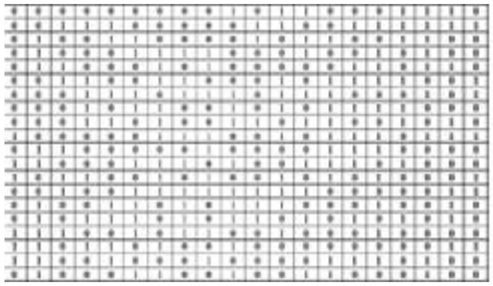
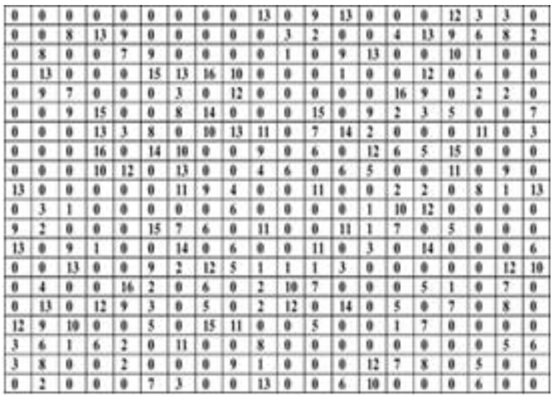


***Fig2:** the matrix formed from the first stage*

2: The connection matrix of the network (8 nodes)



0	0	0	0	0	0	0	0	0	13	0	9	13	0	0	0	12	3	3	0
0	0	8	13	9	0	0	0	0	0	3	2	0	0	4	13	9	6	8	2
0	8	0	0	7	9	0	0	0	0	1	0	9	13	0	0	10	1	0	0
0	13	0	0	0	15	13	16	10	0	0	0	1	0	0	12	0	6	0	0
0	9	7	0	0	0	3	0	12	0	0	0	0	0	16	9	0	2	2	0
0	0	9	15	0	0	8	14	0	0	0	15	0	9	2	3	5	0	0	7
0	0	0	13	3	8	0	10	13	11	0	7	14	2	0	0	0	11	0	3
0	0	0	16	0	14	10	0	0	9	0	6	0	12	6	5	15	0	0	0
0	0	0	10	12	0	13	0	0	4	6	0	6	5	0	0	11	0	9	0
13	0	0	0	0	0	11	9	4	0	0	11	0	0	2	2	0	8	1	13
0	3	1	0	0	0	0	0	6	0	0	0	0	1	10	12	0	0	0	0
9	2	0	0	0	15	7	6	0	11	0	0	11	1	7	0	5	0	0	0
13	0	9	1	0	0	14	0	6	0	0	11	0	3	0	14	0	0	0	6
0	0	13	0	0	9	2	12	5	1	1	1	3	0	0	0	0	0	12	10
0	4	0	0	16	2	0	6	0	2	10	7	0	0	0	5	1	0	7	0
0	13	0	12	9	3	0	5	0	2	12	0	14	0	5	0	7	0	8	0
12	9	10	0	0	5	0	15	11	0	0	5	0	0	1	7	0	0	0	0
3	6	1	6	2	0	11	0	0	8	0	0	0	0	0	0	0	0	5	6
3	8	0	0	2	0	0	0	9	1	0	0	0	12	7	8	0	5	0	0
0	2	0	0	0	7	3	0	0	13	0	0	6	10	0	0	0	6	0	0

***Fig3:** the bandwidth values of the given network*

Third: depending on the second stage the number of the paths will be
No. paths = 8
Fourth: specify the arcs for each path:
Path1 (1-2), (2-8)
Path2 (1-3) (3-4) (4-8)
Path3 (1-3) (3-4) (4-2) (2-8)
Path4 (1-3) (3-4) (4-7) (7-8)
Path5 (1-5) (5-6) (6-7) (8-7)
Path6 (1-5) (5-6) (6-4) (4-8)
Path7 (1-6) (6-4) (4-8)
Path8 (1-6) (6-7) (7-8)
Fifth: calculate the value of each path and specifying the value of each path:
Path1=30.
Path2 =31.
Path3 =49.
Path4 =43.
Path5 =47.
Path6 =34.
Path7 =27.
Path8 =30.
The path which holds the least value is path 7, so the shortest path is path 7=27.

4. Conclusion

The proposed algorithms reduce the amount of data entered to the computer, because we used to inter just the number of the start node of the arc and the number of end node of the arc. The proposed Algorithms give the result in least time and very accurate. It reduces the effort and time in finding the shortest path.

References

[1] Fasial Khamayseh,Nabil Arman,"Improvement of Shortest-Path Algorithms Using Subgraphs Heuristics",JATIT,vol.76.NO.1,JUN.2015.

[2] T.H.Cormen,C.E.Leiserson, R.L.Rivest,and C.Stein."Dijkstras's Algorithm.Introduction to Algorithms (second ed.).Section 24.3:pp.595-601.MIT Press and MC Graw Hill.2001.ISN 0-262-03293-7.

[3] N.Arman, "An Efficient Algorithm for Checking Path Existence Between Graph Vertices", Proceeding of the 6th international Arab Conference on Information Technology (ACIT, 2005), pp.471-476, December 6-8.2005, Al-Isra Private University, Amman,Jurdan,2005.

[4] David Eppstein "Finding the K Shortest Path", University of California ,http://www.Ics.uci.edu/eppsteien/. March 31, 1997.

[5] Husain Ajazzar, Stefan Leue,"K*: A heuristic Search Algorithm for Finding the K Shortest Paths". Artificial Intelligence, 174 (2011)212-2154.

[6] Ahmad Younes Hmed ,"A Genetic algorithms for finding the K Shortest path's in a Network", Egypt Informatics Journal (2010) 11,75-79.

[7] A.W.Brander and M.C.Sinclair, "A Comparative Study of K Shortest Path Algorithms". Proc.11th UK performance Engineering Worsh. For computer and Telecommunications Systems, September, 1995.

[8] R.K.Ahuja,K.Mehlhorn,J.B.Orline, and R.E.Tarjan,"Faster Algorithms for the Shortest Path Problem",J.Assoc compute.Mach.37:213-223.Assoc.forcomputing Machinary,1990.

[9] T.H.Byers and M.S.Waterman,"Determining all Optimal and Near-Optimal Solutions When Solving Shortest Path Problems by Dynamic Programming". Operations Research 32:1381-1384,1984.

[10] Marius Glabowski, Bartoz Musznicki, Przemyslaw Nowrk, "Shortest Path Problem Solving Based on Ant colony Optimization Meta heuristic", image Processing &Communication, Vol.17,no.1-2,PP.7-18. DOI:10.2478/V10248-012-0011-5

[11] P.N.Klen, S.Rao, M.H.Rauch, and S.Subramnian."Faster Shortest-path Algorithms for Planner Graph".Proc.26th Symp. Theory of Computing.PP.27-37.Assoc.for Computing Machinary, 1994.