Journal of Engineering and Development

www.jead.org
Vol.19, No. 06, November 2015
ISSN 1813-7822

DESIGN AND IMPLEMENTATION OF SHORTEST PATH ALGORITHM FOR NETWORK OF ROADS

Nadia Moqbel Hassan*

Assist.Lecturer, Computers & Software Eng. Department, Al-Mustansiriyah University, Baghdad, Iraq.
(Received: 22/2/2015; Accepted: 14/4/2015)

Abstract: In this paper a new method programmed computerizing has been suggested which depends on converting the roads network to matrix and the representation of this matrix in adjacency matrix, that a number of rows is equal number of columns, where the degree of this matrix is equal for number of the edges plus one in (i) multiplied by number of the edges plus one in (j) of the roads network (graph). The elements of This matrix will be either (1) that means existing edge of orientation in one direction connecting between two nodes continuous connection, or (0) and that means there is no edge of oriented connecting between two nodes. And this converting has been implemented by using new algorithm characterized by finding the shortest path between two nodes in the roads network ,whatever its size, type and complication degree, in specific lengths, costs and terms and for more accuracy, to implement this algorithm, the matrix has been processed by three basic steps: The first step is to know and determine a number of the paths in the network and it is the first data which must be input to the computer in running the program. The second step is to specify the edges within installation, each path and its numbers. And the third step is to compare between the lengths of the calculated paths to find the shortest path. The results of this proposed algorithm have shown that it has very high efficiency to get very accurate results. The algorithm has been proven to reduce the size of the input data to the computer as well as to the time and effort to find the shortest path between two nodes in the designed network.

Keywords: *shortest path algorithm, graph algorithm, network, transportation, and graph theory.*

تصميم وتنفيذ خوارزمية اقصر مسار لشبكة الطرق

الخلاصة: في هذا البحث تم اقتراح طريقة جديدة مبرمجة حاسوبيا تعتمد على تحويل شبكة الطرق (Network roads) إلى مصفوفة ثنائية مربعة (Matrix) وتمثيل هذه المصفوفة بطريقة مصفوفة التجاور (Adjacency Matrix)، أي إن عدد الصفوف مساوي لعدد ألأعمدة حيث إن درجة هذه المصفوفة (Degree) مساوي لعدد الحافات +١ في (i) مضروبا في عدد الحافات +١ في (j) في شبكة الطرق أي (Graph). وعناصر هذه المصفوفة تكون إما (١) وهذا يعني وجود حافة موجه باتجاه واحد تربط بين نقطتين ارتباط متواصل أو (٠) وهذا يعني انه لا توجد حافة موجه تربط بين نقطتين. وقد تم تنفيذ هذا التحويل باستخدام خوارزمية جديدة والتي تتميز بإيجاد اقصر الطرق بين نقطتين في شبكة الطرق،مهما كان حجمها ونوعها ومهما كانت درجة تقعيدها بأطوال وكلف وفترة زمنية معينة. ولزيادة دقة تنفيذ هذه الخوارزمية تمت معالجة المصفوفة بثلاث خطوات أساسيه الخطوة الأولى هي لمعرفة وتحديد عدد المسارات في الشبكة وهي أولى البيانات التي يجب إدخالها للحاسوب عند تشغيل البرنامج. والخطوة الثانية هي لتحديد الحافات الداخلة في تركيب كل مسار وعددها. والخطوة الثالثة للمقارنة بين أطوال المسارات المحسوبة في هذه الخطوة لتحديد اقصر مسار. وقد أظهرت النتائج لهذه الخوارزمية المقترحة بأنها ذات كفاءة عالية جدا في الحصول على النتائج بدقة متناهية. وأثبتت الخوارزمية تقليل حجم البيانات المدخلة إلى الحاسوب بالإضافة إلى تقليل الوقت والجهد لإيجاد اقصر الطرق بين نقطتين في الشبكة المصممة.

* nadiamoh_2007@yahoo.com

1. Introduction

Science increasing advanced and developed whenever process mathematically modeling theories and laws. A graph is a mathematical structure that models pairwise relationships among items of a certain form. The abstraction of graphs often greatly simplifies the formulation, analysis, and solution of a problem. Graph theory has many applications. For example, we can use graphs to show network of roads or how different chemicals are related or to show airline routes. Graphs also have be used to show the highway structure of a city, a state, or a country. The edges connecting two nodes can be assigned a non-negative real number, called the weight of the edge. If the graph represents a network of roads, the weight can represent the distance between two places or the travel time from one place to another. Such graphs are called weighted graphs. A graph is a mathematical structure that models pairwise relationships among items of a certain form[1,2,3].To write programs that process and manipulate graphs, the graphs must be stored that is, represented in a computer memory. A graph can be represented (in a computer memory) in several ways. Now we are discussing one of commonly used ways: adjacency matrices and adjacency lists. A simple representation is given by an adjacency matrix, thus the graphs are one of the most used models for realistic solutions of the problems by using computer [2,3,4]. This type of problems presupposes the existence of nodes, its number (m) connects between them through a network from transportation roads (edges connect between pairs of nodes), and it suppose that for every edge non- negative value (C_{ij}) (this value is the weight) and the non-negative value can express transportation cost between two nodes or the distance between two nodes or the transferring time between two nodes.

In order to characterize between nodes [5], we take:

1. The primary node(source)or node of the launching, and it is symbolized by (S) and will be numbered as number (1).
2. The final node (terminal) or node of access, and it is symbolized by (T) and will be numbered as number (m).
3. The other nodes are middle nodes and we will give them numbers (2,3,….m-1).

The edges connecting between the middle nodes can be possible twin the direction, if we have two nodes (i) and (j) the node (i) can transfer to (j) and vice versa, it means one edge has two non- negative values, but everyone in direction. And if there is an edge that has one direction only, then consider that the value of the other direction is zero, but the edges launched from the primary node, and the edges which are accessible for node must be connected in one direction.

This suppositions requires to find the shortest path (according to the nature of the problem) to move from the node(S) to the node (T**)** through middle nodes, and accessing between them by network of the roads [1,4].

2. Literature Survey

In reality there are many styles or algorithms that can be applied to find the shortest path between two nodes connecting between them a series of paths [1, 3, 4, 5], depending on resolving problems in shortest path:-

1. Direct solution on the graph:
   These algorithms are possible when the network of roads is small, and that means the number of nodes is very little and its edges have one direction [1, 4].
2. Dynamic programming: This need some modifications of the problem to fit the dynamic programming style [6].

All these algorithms are possible for small size network, its edges are relatively little provided that it's edges must be directed, that means one edge connected from one side only[7,8]. And in spite of these terms and as result of resolving a number of problems we noticed that the more network increased the more the mathematical processes increased as a result of applying these methods or algorithms, and we need more time and energy in application [9, 10]. Either the networks, which some of its edges have doubled direction, these methods and algorithms will become almost impossible, and if we found the answer, it will be very complex, and it will consume great time and energy [1,2,11].

Otherwise and from a computerized perspective all the methods or algorithms are so hard to account and to put a comprehensive computerized program for all cases and that prevents us from the structuring of networks especially these algorithms and methods which depend on finding the answer on the graph directly [1,4,12,13,14].

Hence the importance of this paper lies in designing and implementing new proposed programmable algorithms, whatever the size of roads network of the problem under study.

This algorithm achieved many goals, together the most important are:

1. Making inspection process through the shortest mechanical path by programmable computerizing.
2. Processing the networks that have doubled edges of direction.
3. Processing the networks, whatever the number of the edges connecting node the launching and node of access.

3. Idea of Conventional Algorithm

The idea of algorithm depends on converting the roads network to the matrix that its columns and rows number are equal to number of edges of the network, and its elements are: either (1) which means that there is an edge connecting the two nodes, or (0) which means that there is no edge connecting the two nodes, and then processing this matrix in three basic stages:

1. The stage of identifying the number of paths in the network by connecting a number of the elements which equal (1) in the first row from the matrix that represents the initial number of paths, then moving to the following rows and counting the number of the elements which equal (1), if its number was more than one at that time other paths will branch out of this row,

its number equals the elements numbers which equal (1) minus one, and this number will be added to the initial number of the paths, and so on.

2. The stage of limiting the number of edges, which is embedded in each path by identifying the number of the row and column of the elements that is equal (1).
3. The stage of accounting the lengths of the paths according to the edges embedded in each path and comparing them to determine the shortest one [1,4].

By using the computer, in running a program it will be enough to input a number of edges in the network, and the number of start of every edge and its end also, as well as the value of the edge (cost or time or length), and the computer processing matrix network which designed above through limiting the possible paths in the network, however it was complicated, and the edges which has been input in every path, and lengths of paths, then select the shortest one [1].

4. roposePd algorithm

The finding of the shortest path between the nodes (1) and (m) connected by a network of roads follows the following stages:

Stage 1: Do the following:

1**.** Identify a number of the edges in the network, for example (n).

2. Punctuate edges of the network by sequential numbers beginning from (1) to (n).

3. The edge (i) in the network connects between two nodes, and according to the necessity of the computerizing programming it will be symbolized as the following:

a) For node number start of the edge (i), symbolized. **Start (i, 1)**.

b) For the node of end the edge (i), symbolizes this. End (i, 2). Where i=1,2,3,…,n. We will symbolize the value of edge by C (i), while the value can be distance or time or cost according to the nature of the problem under study.

And in order to implement the following stages for the algorithm computerizing:

1. Adding an edge to the start of the network which takes number (0) and number of the start node (0) that means: Start (0, 1)=0.

And the number of its end node will take the same number of node for the start of edge

(1) that means in the network: End (0, 2) = Start (1, 1) =1.

2. Adding an edge to the end of the network takes the number (n+1) and the number of its start node takes the same number of its end node for edge (n) in the network. That means: Start(n+1,1)= End(n,2)=m and number of node its end m+1 that means:

End (n+1,2)= m+1

That means an edge is added to the start of the network and an edge of its end, and that is very necessary to implement the following computerizing stages.

Stage 2: In the second stage a matrix(X) is formed and its degree from the level [n+1][n+1], its columns numbers from (1) till (n+1), its rows numbered from (0) till (n) and its elements account as the following: x(i, j)=1.

If the number of the node for the end of edge was (i) equal to the number of the node for the start of edge (j), that means to put (1) if the edge (j) follows the edge (i) directly: and x(i, j)=0 excepts that: i=0,1,2,3,…..,n and j=1,2,3,….n+1.

Stage 3: In the third stage we identify a number of paths in the network.

We symbolize by (path) for the number of paths in the network, and by row(i) for number of elements in the row i, which its value equals one, to identify the number of paths in the network(computationally), the following steps are followed:

Step 1: Path=0, is put.

Step 2: The beginning will be from the first row, that means i=0 and row(0)=0, is put then the number of elements is accounted which their value equal one, and at this time becomes a value of the path and row(0) in the same number which equal the one in the first row and that means:

IF x(i, j)=1 ; j=i+1, i+2, i+3,…..,n+1.
Then row (i)= row(i)+1,path=path+1.

Step 3: The movement in this step will be to the next row i=i+1 and put row(i)=0, then we account the number of the elements which their value equal one in this row, then the value of row(i) becomes equal to the number of elements which equal one, and that means:

IF x(i, j)=1 ; j=i+1, i+2, i+3,…..,n+1.
Then row (i)= row(i)+1.

Step 4: If row (i) was > 1it will be number of the paths in the network that will increase in amount of the elements equal one minus one and that means:

IF row (i) > 1 THEN path=path + (row (i)-1)
And that corresponds on the network, that means there are more than edge kicks off from the node **End (i.2)**.

Step 5: Both steps (3) and (4) are repeated until they become i=n and at that time the number of the edge will be limited which come after every edge i through row(i) and the number of the paths will be accounted in the network through the value of the path.

Stage 4: In this stage the edges are identified involved in composing every path and its number. For the necessity of computationally programming we will use the following symbols:

(k) the number of the activities in every path.
(m) the number of the path.
(L2) the storing of the column.
(Y2) the storing of the row.
(Y2) for storing the number of the column (and for coming back to the element, it has been processed in the former stage).
(S(m, k)) for storing the number of activity involved in composing the path.

In order to identify the edges involved in composing every path and its number, the following steps are used:

Step1: (m=1) and (k=0) are put.

Step2: The beginning will be from the first row (i=0) and the first element x (0, j)

Is taken as equal one, where j= 1, 2, 3, ……, n+1, and we keep its column and row, and Y1= i, Y2=j, L2=j are put.

Then (1) for the chosen element is replaced by (0), that means we put x(0, j) =0 and move to the next step 3. But if all the elements where in the first row equal (0) we will stop tracking, and that we will limit all the possible paths in the network, then we turn to the fifth stage.

Step3: The movement now will be to the row which its column has limit in the former step, that means:

row(i) =1. And here two situations appear:

The first situation: If the number of elements which equal (1) in the row equal (1), that means: row (i) =1, at this time:

1. Identify the only element column equal (1), and that means: L2= j.
2. k = k + 1 and S(m, k) =i are put, and that means: edge (i) enters in composition of the path.
3. If L2 = n + 1, at this time, the movement will be to the step (4) and if not we must come back to the step (3).

The second situation: If there was more than one branched edge, that means:

row (i) > 1, at this time:

1. Identify the first element column which equal (1) in this row, that means: L2 = j.
2. k = k + 1 and S(m, k) =i are put, that means: edge (i) enters in composition of the path.
3. x(i, j)= 0 is put, and that means we make value of the chosen element equal (0).
4. row(i)= row (i) -1 are put, and that means we will minus the number of the elements in this row number one.
5. The coming back will be to the element x(Y1, Y2) that is equal (0) in the former stage, and make it equal one and that means: x(Y1, Y2) =1.
6. We put: row(Y1)= (Y1)+1and that means number of the element will increase in row Y1number one.
7. Number of column and row of the element are kept which we that is equal (0) in number (**3.**)from the second situation, and that means: Y1=I, Y2=j.

If it was L2= n + 1at that time we move to the step (4), if not we will come back to the step (3).

Step4: 1. path (m) = k is put for storing the number of the edges in the path m.

2. m= m + 1is put to identify the edge of the next path.

3. k= 0 is put, then the coming back will be to the step (2) from the fourth stage.

Stage 5: This stage, identify the shortest path there are two steps:

Step1: Identify the lengths of the paths (distance or cost or time), by using equation(1)**:**

$$\text{Total\_Path} = \sum_{k=1}^{\text{path(m)}} C(i) * \big(S(m,k)\big) \qquad (1)$$

Where: m = 1,2,3,……, path.

Step2: In the second stage, identify the shortest path by using the equation (2):

$$\text{Shortest}_{\text{Path}} = \underset{m=1,2,2,\ldots\ldots\ldots,\text{path}}{\text{Min}} \left[\text{Total}_{\text{path(m)}}\right] \qquad (2)$$

5. Simulation Results

Simulation is the implementation to test the performance of algorithm. And to illustrate the results of the proposed algorithm mentioned above, let us take the network or graph as shown in figure (1), to explain the stages of the proposed algorithm:

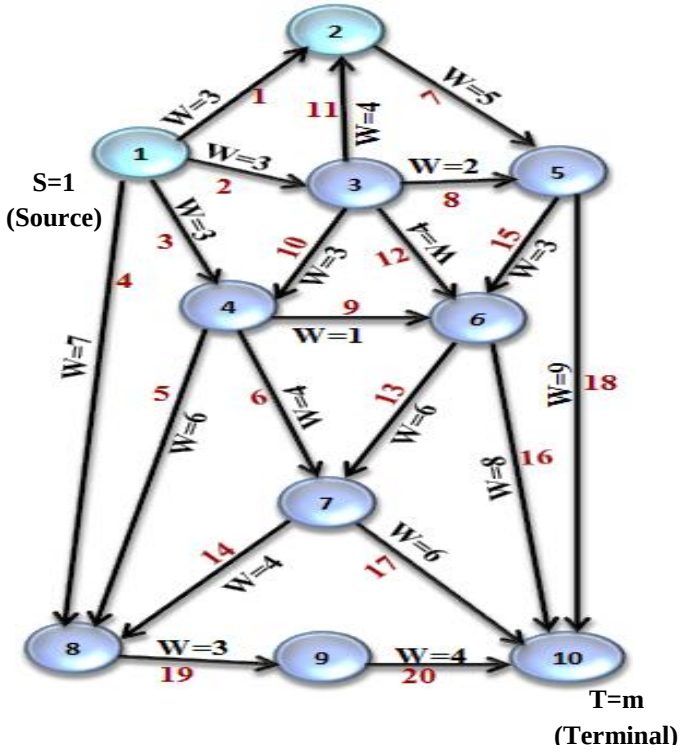

Figure 1. Weighted graph

First: According to the first stage, the data which must enter the computer when the program run it are:

1. The number of the network edges are (20) edges.
2. The number of the starting every edge and its end, and value of every edge as it shown in the following table (1).

As we mentioned formerly, to implement the next stages of the algorithm, must be added an edge to the beginning of the network and an edge to its end and the value of every edge (0), as it is shown in figure (2) and the table (2).

Table 1. The number of the starting every edge and its end, and the value of every edge in the designed network

Edges	Start of edges(i,1)	End of edges (i,2)	Value of edges c(i)
i=1	١	٢	٣
i=2	١	٣	٣
i=3	١	٤	٣
i=4	١	٨	٧
i=5	٤	٨	٦
i=6	٤	٧	٤
i=7	2	٥	٥
i=8	٣	٥	٢
i=9	٤	٦	١
i=10	٣	٤	٣
i=11	٣	٢	٤
i=12	٣	٦	٤
i=13	٦	٧	٦
i=14	٧	٨	٤
i=15	٥	٦	٣
i=16	٦	١٠	٨
i=17	٧	١٠	٦
i=18	٥	١٠	٩
i=19	٨	٩	٣
i=20	٩	١٠	٤

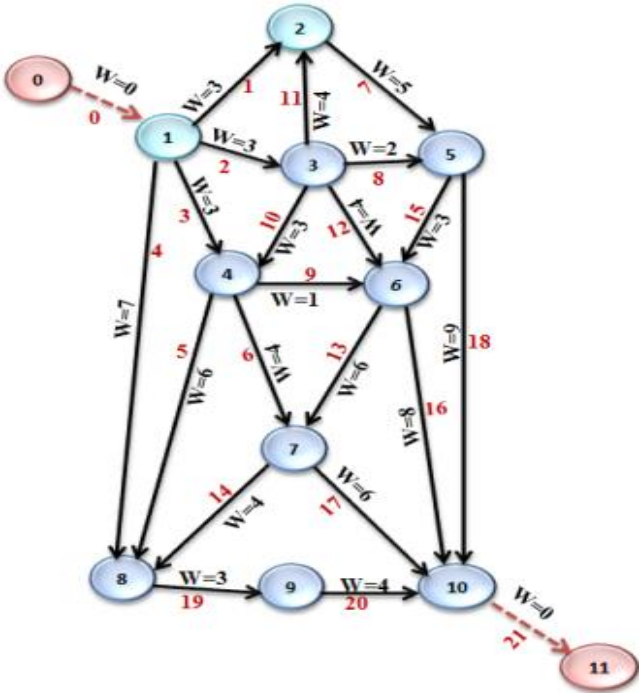

Figure 2. Weighted graph (After adding the edges)

Table 2. The number of the starting every edge and its end, and the value of everyedge in the designed network(After adding the edges)

Edges	Start of edges(i,1)	End of edges (i,2)	Value of edges c(i)
i=0	٠	١	٠
i=1	١	٢	٣
i=2	١	٣	٣
i=3	١	٤	٣
i=4	١	٨	٧
i=5	٤	٨	٦
i=6	٤	٧	٤
i=7	2	٥	٥
i=8	٣	٥	٢
i=9	٤	٦	١
i=10	٣	٤	٣
i=11	٣	٢	٤
i=12	٣	٦	٤
i=13	٦	٧	٦
i=14	٧	٨	٤
i=15	٥	٦	٣
i=16	٦	١٠	٨
i=17	٧	١٠	٦
i=18	٥	١٠	٩
i=19	٨	٩	٣
i=20	٩	١٠	٤
i=21	١٠	١١	٠

Second: According to the second stage, the computer will operate the following matrix (X)

i= \ j=	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
10	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Third: According to the third stage, it identify the number of the paths in the network:
Number paths in network = 18

Fourth: According to the fourth stage, it identify the involved edges in composing of every path:

Path (1)=(1,2),(2,5),(5,6),(6,7),(7,8),(8,9),(9,10).
Path (2)=(1,2),(2,5),(5,6),(6,7),(7,10).
Path (3)=(1,2),(2,5),(5,6),(6,10).
Path (4)=(1,2),(2,5),(5,10).
Path (5)=(1,3),(3,5),(5,6),(6,10).
Path (6)=(1,3),(3,5),(5,10).
Path (7)=(1,3),(3,4),(4,8),(8,9),(9,10).
Path (8)=(1,3),(3,4),(4,7),(7,8),(8,9),(9,10).
Path (9)=(1,3),(3,4),(4,7),(7,10).
Path (10)=(1,3),(3,4),(4,6),(6,7),(7,10).
Path (11)=(1,3),(3,4),(4,6),(6,10).
Path (12)=(1,3),(3,2),(2,5),(5,10).
Path (13)=(1,3),(3,6),(6,7),(7,10).
Path (14)=(1,3),(3,6),(6,10).
Path (15)=(1,4),(4,8),(8,9),(9,10).
Path (16)=(1,4),(4,7),(7,10).
Path (17)=(1,4),(4,6),(6,10).
Path (18)=(1,8),(8,9),(9,10).

Fifth: According to the fourth stage, and as it is shown in figure (3) it will account the length of every path and identifying the shortest path in the designed network.

Total path(1)= 28
Total path(2)= 23

Total path(3)= 19
Total path(4)= 17
Total path(5)= 16
Total path(6)= 14
Total path(7)= 19
Total path(8)= 21
Total path(9)= 16
Total path(10)= 19
Total path(11)= 15
Total path(12)= 21
Total path(13)= 19
Total path(14)= 15
Total path(15)= 16
Total path(16)= 13
Total path(17)= 12
Total path(18)= 14

Shortest path is path (17) = 12.

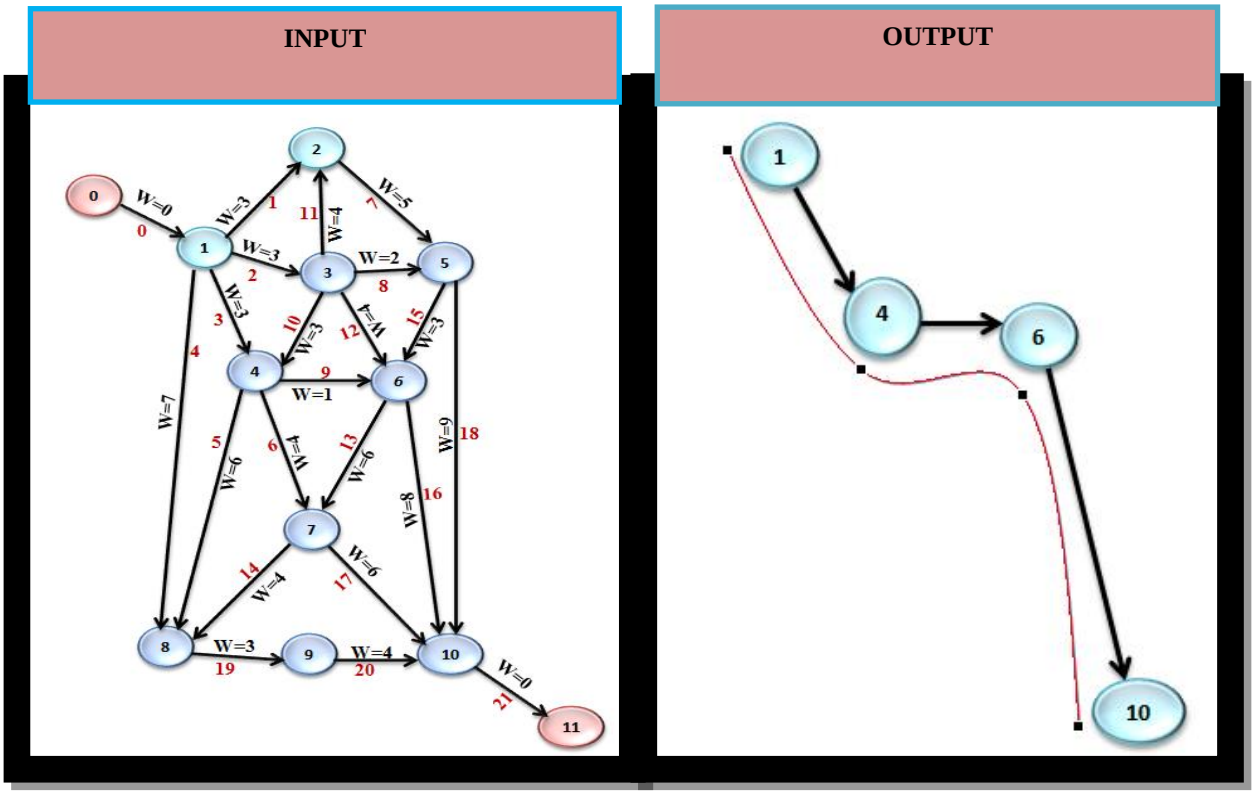

Figure 3. Input network for roads, and output for shortest path Algorithm

6. Conclusion

1. The suggested computer algorithm which it's reached there to in this research decreases the size of the data input into the computer, that only a number of network

edges and the beginning of each edge, it's end and value are input (the value as it is above mentioned either it is a distance or a cost or a time) and the computer punctuated the network edges spontaneously and transforms them into a suitable matrix, then, they are treated according to the algorithms steps and stages.

2. The same algorithm in the first point(1) in this research gives results in high speed and very accurately by inputting the data are required to be input in their correct form.
3. In comparison with other algorithms above mentioned, the suggested computer algorithm decreases the effort and time in finding the shortest path between two nodes or centers connected by a network of roads.
4. The suggested algorithm is recommended in many among them:
   a. How different chemicals are related.
   b. Airline routes.
   c. Highway structure of a city, state, or country.
   d. Project scheduling in studying the networks of works concerning in time or cost.

7. Reference

1. D.S. Malik, 2010. " *Data Structures Using C++* " ,Second Edition, USA.
2. Chung-Yang, Chao-Yue Lai, Kwang-Ting," *Fundamentals of algorithms*".
3. T. Harju, " *GRAPH THEORY*", Department of Mathematics University of Turku FIN- 20014 Turku, Finlan1994 – 2011.
4. A. Drozdek, 2001. " *Data Structures and algorithms in C++*", Second Edition.
5. Abdul-Q. Alzalloum, 2009. "*Application of shortest path algorithms to find paths of minimum radiation dose*", University of Illinois at Urbana-Champaign.
6. B. Doerr, A. Eremeev, F. Neumann, M. Theile, C.Thyssen, (2013)," *Evolutionary Algorithms and Dynamic Programming*", arXiv:1301.4096v1 [cs.NE] 17 Jan.
7. ZHANG Chi-jun,YANG Yong-jian,ZHAO Hong-bo.(2006),"*Improvementand Realiz ation of the Shortest Path Algorithm Based on Path-dependent. Computer Engineering and applications*".,25(2) ,pp.56-58.
8. S. Sanan, L. jain, B. Kappor,(2013), " *Shortest Path Algorithm*" , International Journal of Application or Innovation in Engineering & Management (IJAIEM).Web www.ijaiem.org Volume 2, Issue 7, July .
9. Liu Xiao-Yan, Chen Yan-Li,(2010),"*Application of Dijkstra Algorithm in Logistics Distribution Lines*", Proceedings of the Third International Symposium on Computer Science and Computational Technology(ISCSCT '10) 14- 15,August , pp. 048-050.
10. S. Pallottino, M. Grazia,(1999)" *Shortest Path Algorithms in Transportation Models Models: Classical and Innovative Aspects* ".
11. S.Sahue,(2013)"*PROGRAMMING 201(DATA STRUCTURES USING C++)*", FACULTY OF MEDIA, INFORMATION AND COMMUNICATION TECHNOLOGY,.
12. M.V.Mawale, Dr. Y.B.Gandole,(2011)," *Analysis of Optimal Route Algorithms UnderConstraint Conditions*", M.V. Mawale et al / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (6) , 2614-2619.
13. F. Benjamin Zhan,(1997)" *Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures*", Journal of Geographic Information and Decision Analysis, vol.1, no.1, pp. 70-82,.
14. J. Kent,(2004), " *C++ Demystified:* ", McGraw-Hill/Osborne.