International Journal of Mathematical Archive-9(6), 2018, 173-178
Available online through www.ijma.info ISSN 2229 – 5046

SHORTEST PATH OF A DOUBLY WEIGHTED GRAPH IN TRANSPORTATON PROBLEMS

ARUN KUMAR BARUAH[1] AND MANOSHI KOTOKY*[2]

[1,2]Department of Mathematics,
Dibrugarh University, Dibrugarh - 786004, Assam, India.

(Received On: 08-05-18; Revised & Accepted On: 08-06-18)

ABSTRACT

One of the important issues in everyday life is optimization problem involving reduction of cost and distance of distribution and transportation of goods. Researchers are always in line with this objective, by providing search tools trying various approaches to minimize such costs and distances. The purpose of this paper is to examine the problem and its solution by using graph theoretic algorithms and related theorems. Also, using the properties of a doubly weighted graph - a graph in which both vertices and edges are weighted, we develop a doubly weighted structure in matters of transport networks and then to find a shortest path with maximum vertex weights.

Keywords: *Vertex weighted graph, doubly weighted graph, digraph, shortest path, transportation, Dijkstra's Algorithm.*

1. INTRODUCTION

During the last decades, graph theory has attracted the attention of many researchers. Graph theory has provided a very nice atmosphere for research of provable techniques in discrete mathematics for researchers. Moreover, many applications in the computing, industrial, natural and social sciences are studied by graph theory.

The shortest path problem is a problem of finding the shortest path or route from a starting point to a final destination. Generally, in order to represent the shortest path problem we use graphs. A graph is a mathematical abstr63act object, which contains sets of vertices and edges. Edges connect pairs of vertices. Along the edges of a graph it is possible to walk by moving from one vertex to other vertices. Depending on whether or not one can walk along the edges by both sides or by only one side determines if the graph is an undirected graph or a directed graph. In addition, lengths of edges are often called weights, and the weights are normally used for calculating the shortest path from one point to another point. In the real world it is possible to apply the graph theory to different types of scenarios. For example, in order to represent a map we can use a graph, where vertices represent cities and edges represent routes that connect the cities. If routes are one-way then the graph will be directed; otherwise, it will be undirected. In some cases weight can be assigned to vertex also. Here we use graphs with weights assigned to their vertices also, which represents the number of passengers in a station, amount of traffic, number of goods etc. There exist different types of algorithms that solve the shortest path problems. Some of the most popular conventional shortest path algorithms [3] are: Dijkstra's Algorithm, Floyd-Warshall Algorithm, Bellman-Ford Algorithm, and Genetic Algorithm (GA).

In this paper one problem of transport network is illustrated using Dijkstra's Algorithm, where the network is doubly weighted.

2. SOME DEFINITIONS

2.1 VERTEX WEIGHTED GRAPH

A vertex weighted graph $G = (V, W_V; E, W_E)$ is a graph where W_E be defined such that for all edges e in E, $W_E(e) = 0$. Since all the weights of the edges are zero, it is as if the edges are not weighted at all. Thus we say that graph G is a vertex-weighted graph and its edge weights are not counted when considering weights [5]. Here a graph with four vertices a, b, c, d and their respective vertex weights W_a, W_b, W_c, W_d are shown in **Figure-1** below:

Corresponding Author: Manoshi Kotoky[2]*
[1,2]Department of Mathematics, Dibrugarh University, Dibrugarh - 786004, Assam, India.

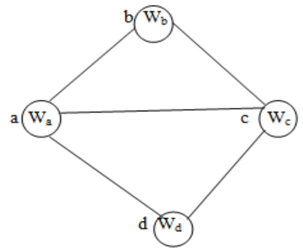

Figure-1: Vertex weighted graph

2.2 EDGE WEIGHTED GRAPH.

An edge weighted graph $G = (V, W_V; E, W_E)$ is a graph where W_V be defined such that for all vertices v in V, $W_V(v) = 0$. Since all the weights of the vertices are zero, it is as if the vertices are not weighted at all. Thus we say that graph G is an edge-weighted graph and its vertex weights are not counted when considering weights [5]. Here a graph with four vertices a, b, c, d and five edges and their respective edge weights W_{ab}, W_{bc}, W_{ac}, $W_{cd} W_{ad}$ is shown in **Figure-2** below:

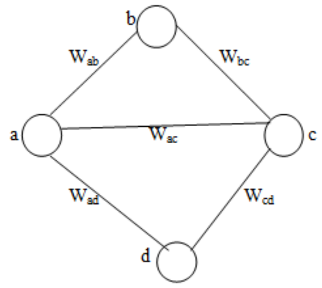

Figure-2: Edge weighted graph

2.3 DOUBLY-WEIGHTED GRAPH

A doubly-weighted graph $G = (V, W_V; E, W_E)$ is a graph where both edges and vertices are weighted. The set V is called the vertex set of G, and elements of this set are called vertices. The vertex weight function $W_V : V \to R$ maps all vertices onto the set of real numbers. We denote the weight of a vertex v as W_V. The set E is called the edge set of G, and elements of this set are called edges. Each edge e_i is defined by two vertices, v_j and v_k, such that $e_i = (v_{j,}, v_k)$. We say that this edge e_i is incident to vertices v_i and v_j. The edge weight function $W_E : E \to R$ maps all edges onto the set of real numbers. We denote the weight of an edge e as $W_E(e)$ [5]. Here a graph with four vertices and five edges and their respective vertex weights and edge weight is shown in **Figure-3** below:

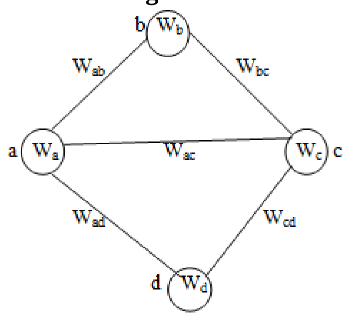

Figure-3: Doubly weighted graph

3. DIJKSTRA'S ALGORITHM

In 1959, Edsger Dijkstra invented an algorithm for finding the shortest path through a network. Dijkstra's algorithm labels the vertices of the given digraph. At each stage in the algorithm some vertices have permanent labels and others temporary labels. The algorithm begins by assigning a permanent label 0 to the starting vertex s and a temporary label ∞ to the remaining $n-1$ vertices. From then on, in each iteration another vertex gets a permanent label, according to the following rules [1];

1. Every vertex j that is not yet permanently labeled gets a new temporary label whose value is given by

   min [old label of j, (old label of $i + d_{ij}$)].

   where i is the latest vertex permanently labeled, in the previous iteration, and d_{ij} is the direct distance between vertices i and j. If i and j are not joined by an edge, then $d_{ij} = \infty$.

2. The smallest value among the entire temporary label is found, and this becomes the permanent label of the corresponding vertex. In case of a tie, select anyone of the vertex for permanent label.

STEPS OF DIJKSTRA'S ALGORITHM

The following is a simple set of instructions to follow the algorithm [4]:
Step-1: Label the start vertex as 0 and a temporary label ∞ to the remaining vertices.
Step-2: Box this number (permanent label).
Step-3: Label each vertex that is connected to the start vertex with its distance (temporary label).
Step-4: Box the smallest number.
Step-5: From this vertex, consider the distance to each connected vertex.
Step-6: If a distance is less than a distance already at this vertex, cross out this distance and write in the new distance. If there was no distance at the vertex, write down the new distance.
Step-7: Repeat from step 4 until the destination vertex is boxed.

Dijkstra's algorithm is applicable to digraphs, but we can use it to an undirected graph also. In an undirected graph $d_{ij} = d_{ji}$, we can consider edges of an undirected graph as two oppositely directed edges of the same weight. Thus we can convert the given undirected graph to a digraph (or directed graph).

4. FORMULATION OF THE PROBLEM

Theory of graphs are used as device for modeling and description of real world network systems such transport, water, electricity, internet, work operations schemes in the process of production, construction, etc. Although the content of these schemes differ among themselves, but they have also common features and reflect certain items that are in the relation between each other. So the scheme of transport network might be considered in manufacturing centers, and roads and rail links connected directly to those centers [2]. The following example is designed for the solution of a practical problem to find a doubly weighted graph for a rail network with maximum vertex weights as well as the minimum edge weight. For this case we develop a network model of the transportation problem which is analyzed in detail to minimize the distance. Since Dijkstra's algorithm is applicable to both directed and undirected graphs. So first we consider the directed case and then the undirected case.

Let us consider a rail road network with ten nodes and nineteen edges, where each node and the edges are assigned with a weight. Here nodes represent railway stations and edges represent railway lines connecting them. Weight on each node represents the number of passengers in that station and that of edge represents the distance between the stations in km.

4.1 DIRECTED CASE
Let us consider a directed graph whose vertices be considered as A, B, C, D, E, F, G, H, I, J with weights W_A, W_B, W_C, W_D, W_E, W_F, W_G, W_H, W_I, W_J respectively and edges between the vertices be AB, AC, AD, BD, BE, CD, CF, DE, DG, DF, EG, EH, FG, FI, GH, GI, GJ, HJ, IJ considered with weights (in km) 4, 8, 8, 3, 12, 5, 20, 4, 14, 20, 9, 9, 8, 11,7, 8, 14, 14, 8 respectively which is shown in the **Figure-4** below.

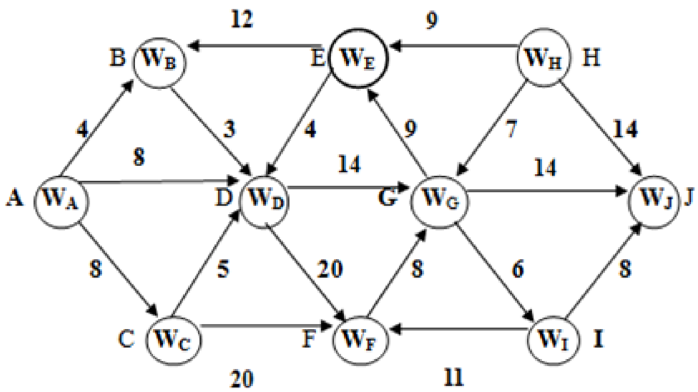

Figure-4: Doubly Weighted Digraph

Using the steps of Dijkstra's algorithm, we can find the shortest path (Minimum distance) between stations *A* and *J*. The permanent labels will be shown enclosed in a square and the most recently assigned permanent label is indicated by a tick which is shown in **Table-1** below

Steps	A	B	C	D	E	F	G	H	I	J
1	$\boxed{0}$ √	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	$\boxed{0}$	4	8	8	∞	∞	∞	∞	∞	∞
3	$\boxed{0}$	$\boxed{4}$	8	8	∞	∞	∞	∞	∞	∞
4	$\boxed{0}$	$\boxed{4}$	8	7	∞	∞	∞	∞	∞	∞
5	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$ √	∞	∞	∞	∞	∞	∞
6	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	∞	27	21	∞	∞	∞
7	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	∞	27	$\boxed{21}$ √	∞	∞	∞
8	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	∞	27	$\boxed{21}$	∞	27	35
9	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	∞	27	$\boxed{21}$	∞	$\boxed{27}$ √	35
10	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	∞	27	$\boxed{21}$	∞	$\boxed{27}$	35
11	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	∞	27	$\boxed{21}$	∞	$\boxed{27}$	$\boxed{35}$ √

Table-1: Labels of vertices of the digraph after using Dijkstra's algorithm

From the above table we find that shortest distance is 35 (km) and the path corresponding to to this distance of 35 are *ABDGJ* and *ABDGIJ*. Now we find the vertex weights of the routes corresponding to *ABDGJ* and *ABDGIJ* and choose the route with maximum vertex weights. The vertex weights of the route *ABDGJ* is $W_A + W_B + W_D + W_G + W_J$ and the vertex weights of *ABDGIJ is* $W_A + W_B + W_D + W_G + W_I + W_J$

If $W_A + W_B + W_D + W_G + W_J > W_A + W_B + W_D + W_G + W_I + W_J$, then we choose the route *ABDGJ*, otherwise we choose *ABDGIJ* so that maximum number of passengers can travel along the shortest path.

4.2 UNDIRECTED CASE

Let us consider an undirected graph whose vertices be considered A, B, C, D, E, F, G, H, I, J with weights W_A, W_B, W_C, W_D, W_E, W_F, W_G, W_H, W_I, W_J respectively and edges between the vertices be AB, AC, AD, BD, BE, CD, CF, DE, DG, DF, EG, EH, FG, FI, GH, GI, GJ, HJ, IJ considered with weights(in km) 4, 8, 8, 3, 12, 5, 20, 4, 14, 20, 9, 9, 8, 11,7, 8, 14, 14, 8 respectively which is shown in the **Figure-5** below.

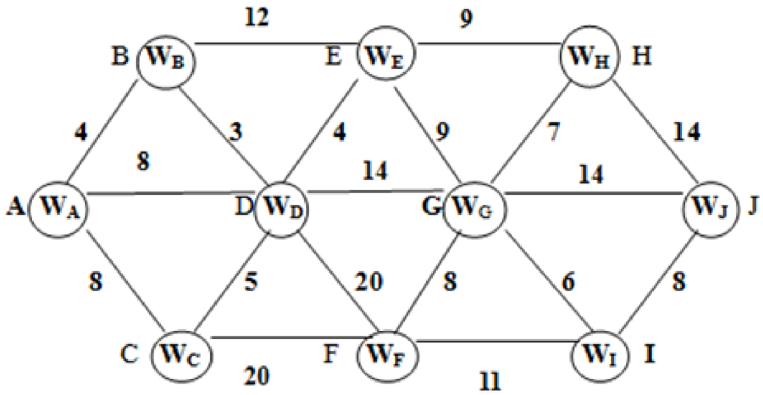

Figure-5: Doubly Weighted Graph

Here we consider an undirected graph. Since the edges of an undirected graph can be considered as two oppositely directed edges of the same weight (mentioned in **section 3**), we can convert the given undirected graph to a digraph (or directed graph).The directed form of the given graph is shown in the **Figure-6** below:

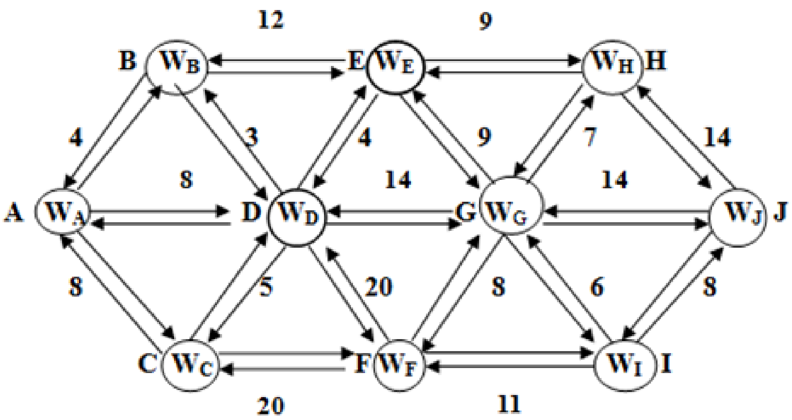

Figure-6: Doubly Weighted Digraph of the undirected graph

Using the steps of Dijkstra's algorithm, we can find the shortest path (Minimum distance) between stations *A* and *J*. The permanent labels will be shown enclosed in a square and the most recently assigned permanent label is indicated by a tick which is shown in **Table-2** below:

Steps	A	B	C	D	E	F	G	H	I	J
1	$\boxed{0}$ √	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	$\boxed{0}$	4	8	8	∞	∞	∞	∞	∞	∞
3	$\boxed{0}$	$\boxed{4}$ √	8	8	∞	∞	∞	∞	∞	∞
4	$\boxed{0}$	$\boxed{4}$	8	7	16	∞	∞	∞	∞	∞
5	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$ √	16	∞	∞	∞	∞	∞
6	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	11	27	21	∞	∞	∞
7	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$ √	27	21	∞	∞	∞
8	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$	27	20	20	∞	∞
There	Is	A	tie	between	vertices	G &H,	first	We	Choose	G
9_1	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$	27	$\boxed{20}$ √	27	26	35
10_1	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$	27	$\boxed{20}$	27	$\boxed{26}$ √	35
11_1	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$	27	$\boxed{20}$	27	$\boxed{26}$	34
12_1	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$	27	$\boxed{20}$	27	$\boxed{26}$	$\boxed{34}$ √

Table-2: Labels of vertices of the undirected graph after using Dijkstra's algorithm and tie breaking at G

Now we choose the vertex H in step 8 of the above table. The steps are given in **Table-3** below:

Steps	A	B	C	D	E	F	G	H	I	J
1	$\boxed{0}$ √	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	$\boxed{0}$	4	8	8	∞	∞	∞	∞	∞	∞
3	$\boxed{0}$	$\boxed{4}$ √	8	8	∞	∞	∞	∞	∞	∞
4	$\boxed{0}$	$\boxed{4}$	8	7	16	∞	∞	∞	∞	∞
5	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$ √	16	∞	∞	∞	∞	∞
6	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	11	27	21	∞	∞	∞

7	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$ √	27	21	∞	∞	∞
8	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$	27	20	20	∞	∞
9_2	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$	27	20	$\boxed{20}$ √	26	35
10_2	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$	27	20	$\boxed{20}$	26	34
11_2	$\boxed{0}$	$\boxed{4}$	8	$\boxed{7}$	$\boxed{11}$	27	20	$\boxed{20}$	26	$\boxed{34}$ √

Table-3: Labels of vertices of the undirected graph after using Dijkstra's algorithm and tie breaking at H

In both the cases, we choose the vertex *J* whose value is not minimum (at steps 12_1 and 11_2), but it is the destination vertex.

From the first table, shortest distance is 34 (km) and the route corresponding to this distance of 34 is *ABDEGIJ*. In the second table shortest distance is also 34 and the route corresponding to this distance of 34 is *ABDEHJ*. In both the cases the shortest distance is 34, now we find the vertex weights of the routes corresponding to case I and case II and choose the route with maximum vertex weights, i.e. in the route corresponding to *ABDEGIJ*, vertex weights is $W_A + W_B + W_D + W_E + W_G + W_I + W_J$ and in the route corresponding to *ABDEHJ*, the vertex weight is $W_A + W_B + W_D + W_E + W_H + W_J$.

If $W_A + W_B + W_D + W_E + W_G + W_I + W_J > W_A + W_B + W_D + W_E + W_H + W_J$, then we choose the route *ABDEGIJ*, otherwise we choose *ABDEHJ* so that maximum number of passengers can travel along the shortest path.

5. CONCLUSION

In the present transportation problem, we have used both vertex weight and edge weight of a graph to minimize the distance between two railway stations with the condition that maximum number of passengers can travel along the shortest path. Here we consider two cases, one is for digraph and the other is for undirected graph. In both the cases we are interested to find the shortest paths with maximum vertex weights. There may be more than one shortest paths if tie condition occurs. In such cases we choose the path whose vertex weights are maximum. For simple networks, where there is only one shortest path, we choose that path and find the vertex weights of that path. Here it is shown that vertices are also important in some cases as the edges. This concept can be used by railway engineers to obtain quantitative data about the passengers and to make their schedule among the stations.

REFERENCES

1. Deo N., Graph Theory with Applications to Engineering and Computer Science, Prentice Hall of India, 2002.
2. Kharazi H., Pourhadi E. Graph Theory in Distribution and Transportation Problems and the Connection to Distance-Balanced Graphs. International Journal Series in Multidisciplinary Research (IJSMR), Vol. 1, No. 3, 2015, 17-30 ISSN: 2455–2461.
3. Magzhan K., Jani H. M., "A Review and Evaluations of Shortest Path Algorithms", International journal of scientific & technology research volume 2, issue 6, june 2013.
4. Shortest path problem (dijkstra's algorithm), chapter 2, https://www.pearsonschoolsandfecolleges.co.uk/.../Chp-02%20023-043.pdf
5. Yu K., Exploring Vertex and Edge Weighted Graphs, bert.stuy.edu/pbrooks/spring2009_old/materials/.../KennyYu1.pdf

Source of support: Nil, Conflict of interest: None Declared.