Shortest distance between two vertices in connected graph using TJ – adjacency of vertices and minimum dominating set

[1]Mr. Tushar J Bhatt, [2]Dr. Jayesh N Zalavadia, [3]Dr. Om. M. Teraiya
[1]Assistant Professor in Mathematics, [2] Assistant Professor in Mathematics, [3] Assistant Professor in Mathematics
[1]Humanities and Science Department,
[1]Atmiya University, Rajkot(Gujarat), India.

Abstract : Domination theory is a part of Graph theory is used for finding communities in networks, device for modeling, description of real world network systems such are : Transport, Water, Electricity, Internet and many more. So using the domination theory and it is mixed with operation research techniques we can try to solve short rout problem. In this paper we can try to find the shortest distance between two vertices using the concept of minimal dominating set mechanism.

Index Terms - Shortest distance between two vertices, Connected graph, Dominating set, Minimal dominating set, minimum dominating set, Domination number, Dijkstra's algorithm.

I. INTRODUCTION

Graph Theory provides many useful applications in operation research. A Graph is defined as finite number of vertices and edges .Mathematically we say that A graph $G = (V, E)$ is a mathematical structure consisting of two sets, a non empty set V known as the vertex set and a set E known as edge set.[2]

- The elements of $V$and E are called vertices and edges respectively.[2]
- Precisely V and E are also denoted as $V(G)$ and $E(G)$ respectively.[2]

A vertex u of graph G is said to be connected to vertex v of a graph G if there is a path from u to v in graph G. In other words a graph G is said to be connected if every pair of vertices are connected.

- Note that a graph G which is not connected is called disconnected.[3]
- Given any vertex u of graph G, $C(u)$ denotes the set of all vertices which are connected to u.[3]

A subset S of $V(G)$ is said to be dominating set if for every vertex v in $V(G) \setminus S$ there is a vertex u in S such that u is adjacent to v.[7]

In other words a subset $S \subseteq V$ is called dominating set if every vertex $v \in V$ is either an element of S or is adjacent to an element of S. [7]

A dominating set S of the graph G is said to be a minimal dominating set if for every vertex v in S, $S \setminus \{v\}$ is not a dominating set.[7]

A dominating set with minimum number of vertices (minimum cardinality of a set) is called minimum dominating set. It is denoted by Υ.

The number of vertices in a minimum dominating set is called domination number of the graph G and it is denoted by $\Upsilon(\mathrm{G})$.[7]

The shortest distance between two vertices is the number of edges in a shortest path.[6]

2. Dijkstra's algorithm [1]

In the Dijkstra's algorithm first we have to find a source vertex or starting point and ending vertex or destination vertex in the graph.

The working rule for Dijkstra's algorithm as follows :

Step -1 : Remove all the loops from the graph.

Step -2 : Remove all parallel edges between two vertices except the one with least weight from the graph.

Step -3 : Create the weight matrix table :

- Set 0 (zero) to the source vertex and infinite to the remaining vertices for all vertices and repeat following
- Mark the smallest unmarked value and mark that vertex in table
- Find those vertices which are directly connected with marked vertex and update all.

Step-4: Update value formula

$$New\ destination\ value = min(old\ destination\ value\ , marked\ value + edge\ weight)$$

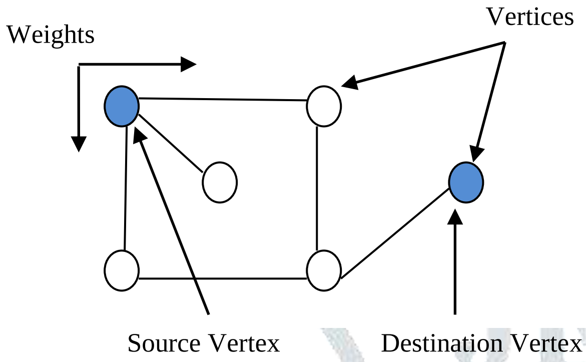

Figure 1

3. Example-1 : As per the following connected graph we can try to find a shortest path between vertex *A* to vertex *F* .[6]

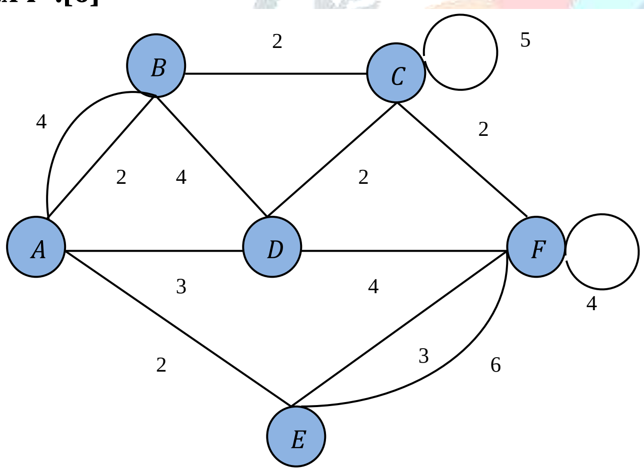

Figure 2

In above figure 2, remove all the parallel edges with maximum weight and loops we get a new figure 3 as follows :

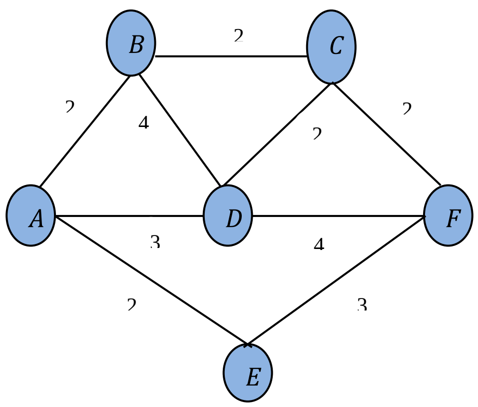

Figure 3

In third step we create the weight matrix (Table-1) in which we want to find the shortest path between two vertices A and F.

Now we can choose a source vertex $\boldsymbol{A}$, the distance between A to A is zero.

Next check the adjacent vertices which are associate with A , showing in figure 3 as

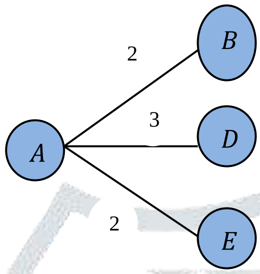

Figure 4

So we can change weights of vertices $B, D,$ and E.
Therefore new destination value of vertex B
$new\ B = \min(\infty, 0+2) = \min(\infty, 2) = 2$
Similarly new destination value of vertex D
$new\ D = \min(\infty, 0+3) = \min(\infty, 3) = 3$
And new destination value of vertex D
$new\ E = \min(\infty, 0+2) = \min(\infty, 2) = 2$
Hence find a minimum value and marked, those marked vertex in column write in the row.
Above procedure is repeated till the destination vertex. The tabular form of above arrangement as follows.

Table – 1

Marked	A	B	C	D	E	F
A	**0**	∞	∞	∞	∞	∞
B		**2**	∞	3	2	∞
E			4	3	**2**	∞
D			4	**3**		5
C			**4**			5
F						**5**

As per the above table – 1 we see that the shortest path between vertex A and vertex F is 5. it means that $A - E - F$.

4. Example-2 : As per the following connected graph we can try to find a shortest path between vertex $\boldsymbol{A}$ to vertex $\boldsymbol{F}$.[6]

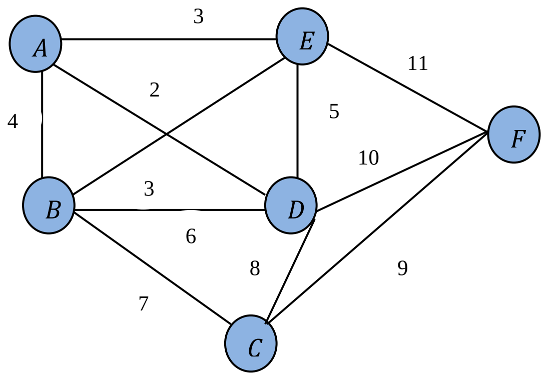

Figure 5

In third step we create the weight matrix (Table-2) in which we want to find the shortest path between two vertices A and F.

First we set zero to the vertex A and according to the step-3 and step-4 we can make a following arrangement in the tabular form.

Table – 2

Marked	A	B	C	D	E	F
A	**0**	∞	∞	∞	∞	∞
D		4	∞	**2**	3	∞
E		4	10		**3**	12
B		**4**	10			12
C			**10**			12
F						**12**

As per the above table – 2 we see that the shortest path between vertex A and vertex F is 12. it means that $A - D - F$.

4. Main Result

In example-1 and example-2 using Dijkstra's algorithm we are able to find the shortest path between two vertices but it is time consuming and lengthy process. So we are trying to define a new little procedure which is immediate give a shortest path between two vertices namely mechanism of TJ – adjacency of vertices as follows :

Conditions :

- The Given graph is connected
- The Source vertex and Destination vertex has common adjacency at least with one vertex.
- The Source vertex and Destination vertex are must be generate the minimum dominating set.

TJ – adjacency Procedure For a given graph there is even number of common vertices

- First find Source vertex and its adjacent vertices with weights
- Next find Destination vertex and its adjacent vertices with weights
- To take the sum of weights of those vertices which are common in both source and end vertices.
- To take a minimum value form the sum for the common vertices weight.
- To take minimum weights of uncommon vertices.
- To take maximum of both the minimum values of weights of common and uncommon vertices. It will give shortest path between the source and destination vertices.

Explanation for example – 1 :

Let us consider example – 1, we want to find a path between the source vertex A and destination vertex F.

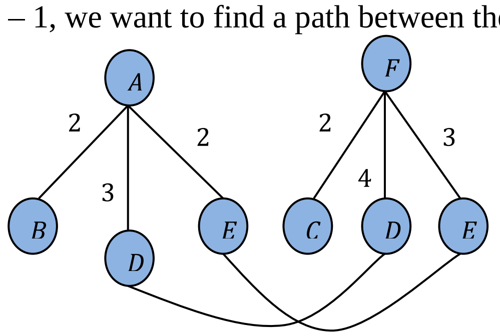

Figure 6

In above figure 6 the Common vertices are D and E.Therefore sum of weights of vertices $D\ and\ D$ and $E\ and\ E$ is 7 and 5.Now take a minimum value form 7 and 5.

$i.e.\min(7,5) = 5$

Uncommon vertices are B and C. Therefore take minimum weights from uncommon vertices is

$i.e.\min(2,2) = 2$

Next we take the maximum weights from common and uncommon vertices is

$i.e.\max(min(7,5), min(2,2)) = \max(5,2) = 5$

So it is verified the shortest path between the source vertex A and destination vertex F is **5.**

TJ – adjacency Procedure For a given graph there is odd number of common vertices :

- First find Source vertex and its adjacent vertices with weights
- Next find Destination vertex and its adjacent vertices with weights
- To take the sum of weights of those vertices which are common in both source and end vertices.
- To take a minimum value form the sum for the common vertices weight.
- To take Maximum weight from source vertex which is uncommon as well as choose Minimum weight from the destination vertex which is uncommon and added together.
- Next to take minimum weight from source vertex which is uncommon as well as choose maximum weight from the destination vertex which is uncommon and added together.
- Next choose the minimum value form above two steps and it is added in minimum weight which is taken from source vertex.
- Hence we get the shortest path between two vertices.

Explanation for example – 2 :

- Let us consider example – 1, we want to find a path between the source vertex A and destination vertex F.

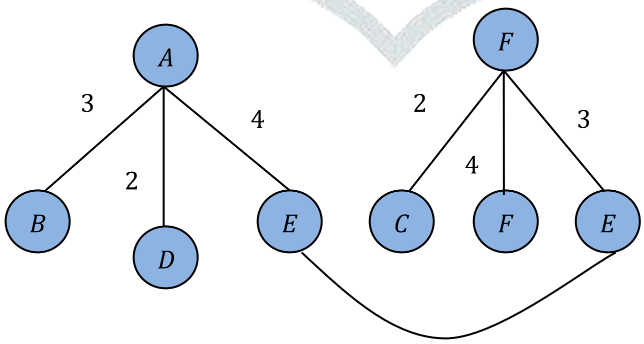

Figure 7

In above figure 7 the Common vertex is E. Therefore sum of weights of vertices $E\ and\ E$ is 7.
$i.e.\min(7) = 7$

Uncommon vertices are B, D, F and C.

Now take maximum weights from source vertex which is uncommon and take minimum weights from destination vertex which is uncommon. And added together.

Maximum weights from source vertex which is uncommon is 3
minimum weights from destination vertex which is uncommon is 2
And addition of both is $3 + 2 = 5$.

Next to take minimum weight from source vertex which is uncommon is 2.
And maximum weight from the destination vertex which is uncommon is 4.
And addition of both is $2 + 4 = 6$.

Now choose the minimum value form above two steps is $\min(5, 6) = 5$

Now the minimum weight which is taken from source vertex is 7.

Now, Shortest path between A to F is
$\min(5, 6) + \min(7) = 5 + 7 = \mathbf{12}$

Concluding Remarks:

TJ – adjacency procedure is useful for quickly find the shortest distance between two vertices if graph satisfying following three conditions :

1. Given graph is connected
2. The Source vertex and Destination vertex has common adjacency at least with one vertex.
3. The Source vertex and Destination vertex are must be generate the minimum dominating set.

TJ – adjacency for a given graph there is even number of common vertices :

➔ Let G be any graph with $n -$ vertices satisfying above three conditions in which marked vertex v_i as a source vertex and v_j as a destination vertex as well as v_i and v_j has even number of common adjacent vertices then the shortest distance between v_i to v_j is given by maximum of "sum of minimum weights associate with common adjacent vertices and sum of minimum weights associate with uncommon vertices".

→ Mathematically we shows that

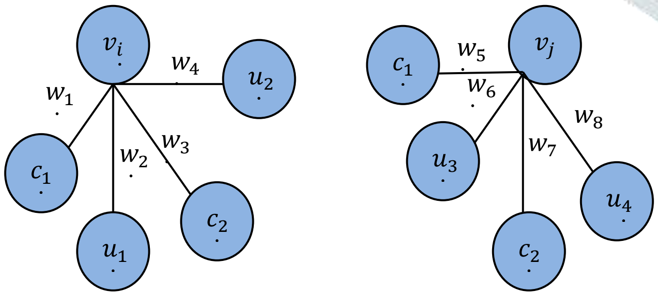

Figure 8

Sum of weights of common vertices $c_1 and\ c_1 = w_1 + w_5 = r_1$.

Sum of weights of common vertices $c_2 and\ c_2 = w_3 + w_7 = r_2$.

Let $S_1 = \min(r_1, r_2)$--------(1)

Now weights of uncommon vertices u_1, u_2, u_3 *and* u_4 are w_2, w_4, w_6 *and* w_8 respectively.

Let $S_2 = \min(w_2, w_4, w_{6,} w_8)$-------------(2)

Now take maximum from result (1) and result (2) we get shortest distance between two vertices v_i *and* $v_j = \max(S_1, S_2)$.

TJ – adjacency for a given graph there is odd number of common vertices :

- Let G be any graph with $n-$ vertices satisfying above three conditions in which marked vertex v_i as a source vertex and v_j as a destination vertex as well as v_i and v_j has odd number of common adjacent vertices then the shortest distance between v_i to v_j is given by minimum of sum of weights of common vertices. Maximum weights from source vertex which is uncommon and take minimum weights from destination vertex which is uncommon and addition of them. Now choose minimum weight from their addition. And again added together both the minimums.
- Mathematically we shows that

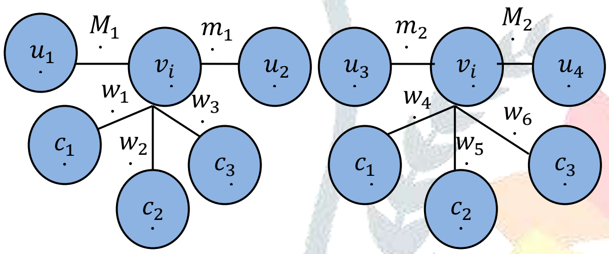

Figure 9

In figure 9 M_1, indicates maximum weight from source vertex to adjacent vertex. m_1, indicates minimum weight from source vertex to adjacent vertex. M_2, indicates maximum weight from destination vertex to adjacent vertex. m_2, indicates minimum weight from destination vertex to adjacent vertex.

Now sum of weights of odd number of common adjacent vertices are,

c_1 *and* $c_1 = w_1 + w_4 = r_1$
c_2 *and* $c_2 = w_2 + w_5 = r_2$
c_3 *and* $c_3 = w_3 + w_6 = r_3$

Now take the minimum weight from $r_1, r_2 and\ r_3$.

Let $r = \min(r_1, r_2, r_3)$

Now take an addition of $M_1 + m_2$ say t_1 and $m_1 + M_2$ say t_2.

Now take minimum from t_1 and t_2 say t.
$i.e. \min(t_1, t_2) = t$

Therefore the shortest distance between two vertices v_i and $v_j = \min(r_1, r_2, r_3) + \min(t_1, t_2) = r + t$.

References :

[1] Ojekudo,Nathaniel Akpofure and Akpan, Nsikan Paul (2017) ‘An Application of Dijkstra’s Algorithm to shortest route problem’, IOSR Journal of Mathematics, Vol. 13, No. 2278-5728, pp.20-32.

[2] J.Clark and D.Holton, A first look at Graph Theory, Word Scientific, 1991(Page No. : 55-125).

[3] F.Harary, Graph Theory, Addison – Wesley, 1969

[4] D.B.West, Introduction to Graph Theory, 2nd Ed., Prentice-Hall of India,2001(Page No. : 11- 58).

[5] R. Balakrishnan, K. Ranganathan, A Textbook of Graph Theory, Springer, New York, 2000(Page No. : 10-39).

[6] Lieberman, F.S, Introduction to Operation Research, 7th Ed. Mc Graw Hill, 2001 (Page No. : 128-223).

[7] Teresa W. Haynes, Stephen T.Hedetniemi, Peter J. Slater, Fundamentals of Domination in Graphs, Marcel Dekker, INC,1998 (Page No. : 1-67).

.