Improved Average Subjective Bell Man Ford Procedure For Coupled Grid

Thiagarajan Kittappa
Professor, Department of Mathematics, Rajalakshmi Institute of Technology, Chennai, vidhyamannan@yahoo.com

Aarthi R
Department of Computer Science & Engineering, K.Ramakrishnan College of Technology, Tiruchirappalli, aarthiravichandranr@gmail.com

Venkatesh. S
Professor, Department of Mathematics, K.Ramakrishnan College of Technology, Tiruchirappalli,Tamil nadu, India, venkateshjjmaths@gmail.com

Anish Anburaja
Department of Information Sciences, University of Maryland, Maryland, USA, anishanbu@gmail.com

Jagathesan M
Department of Computer Science & Engineering, K.Ramakrishnan College of Technology, Tiruchirappalli, jegadeeshmayil007@gmail.com

Abstract

The process of previous algorithms is time-consuming because first, each procedureneeds to explore the entire graph and calculate the shortest path from each node. The proposed modified average weight Bell man ford Procedure gives a more efficient minimum path cost than the well known bellman ford and dijkstra’s algorithm.

Keywords: *average, bellman ford procedure, minimum, negative, shortest path, weight.*

INTRODUCTION

Graph theory and network analysis are one of the most important topics dealt with in operations research. It has been suggested that any system can be represented by a group of nodes, where some of these pairs of nodes are related to specific relationships that we express with lines that connect these pairs of points. This idea led to the emergence of the so-called graph theory. The reason for the development of this theory and its interest in it is due to its applicability in various and varied fields. And given the wide range of problems and the huge and varied number of issues dealt with in graph theory, this theory played and is still playing a big role in dealing with many problems. And the issue of the shortest path is considered one of the most important issues in graph theory.

Research for finding the greedy algorithms for handling the issue of shortest path has been ongoing.In the real world it is easy to apply graph theory to different types of scenarios. In the shortest path algorithm, the study focuses on two nodes or vertices of the path and finds the best solution for the shortest path. Several algorithms are frequently used to discover the shortest path between graph nodes. For example, Bellman-Ford and Dijkstra are the most effective algorithms for single-source shortest path issues. For dense graphs, the Floyd-Warshall is mostly used to discover the shortest path for all pairs and the Johnson procedureis the best for sparse graphs. The process of these algorithms is time-consuming because first, each procedureneeds to explore the entire graph and calculate the shortest path from each node.

This average weight procedure gives a more efficient minimum path than the bellman ford and dijkstra's procedure.

Bellman-Ford Procedure (BMFP) :

Richard Bellman released the Bellman-Ford procedurein 1958 . Bellman-Ford procedureapplied to search the minimum path when some of the edges of the directed graph G may have negative weight from the source node to all other nodes . Graphs with negative edge weights are difficult to resolve by using the procedureof Dijkstra. The procedureof Dijkstra is faster than the BellmanFord algorithm, but more versatile is the Bellman-Ford algorithm. This algorithm, like Dijkstra's Algorithm, uses the idea of area relaxation but doesn't use greedy technique.

The advantages of this procedureare it is a dynamic procedure, it can calculate the negative directed edges (in addition to the positive), can minimize the cost when the network was built, because it can finds the shortest path from one node to another, so we don't have to build a lot of router path. Also, this procedureis simple and it does not need complicated data structures for applications and can find the minimum path weight with high efficiency and accuracy .

The disadvantages of the Bellman procedurewhen used in the Routing Information Protocol (RIP) are that it does not take into account weight and also a slow response to changes in network topology resulting from slow updates passed from the RIP device to the next device. These flaws lead to an attempt to use idle tracks that waste time and network resources.Bellman-Ford procedurereturns a Boolean value to indicate whether a negative cycle can be reached by the origin. If no such cycle occurs, the procedurereturns the shortest path, if a negative cycle exists, the procedurereveals that there is no shortest path> .

The Bellman-Ford procedureis executed as folows

Procedure:

Step 1: Set the distance of source vertex s to zero value (distance[s] = 0) and assign other vertices distance with INFINITY.

Step 2: Relaxes each edge for (n − 1) times when n is the number of nodes. Relaxing an edge means checking to see if it is possible to shorten the route to the node to which the edge points and, if so, replace the route to the node with the route found .

Step 3: Check if the graph has any negative cycle with the Nth loop.

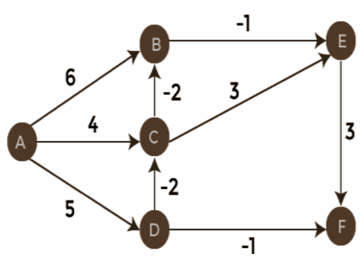
But the time complexity of Bellman-Ford is O(VE), which is more than Dijkstra. where V is a number of vertices and E is a number of edges. For a complete graph with n vertices, V

= n, E = O(n2). So overall time complexity becomes O(n3).

The time complexity of the bellman ford procedurefor the best case is O(E) while average-case and worst-case time complexity are O(NE) where N is the number of vertices and E is the total edges to be relaxed.

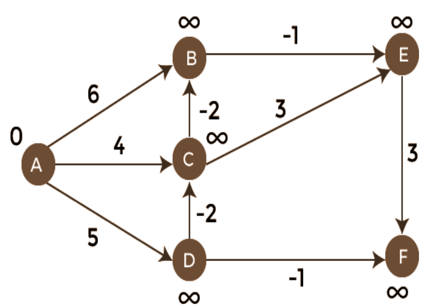
Example 1:

Finding minimum path using bellman ford Procedure:

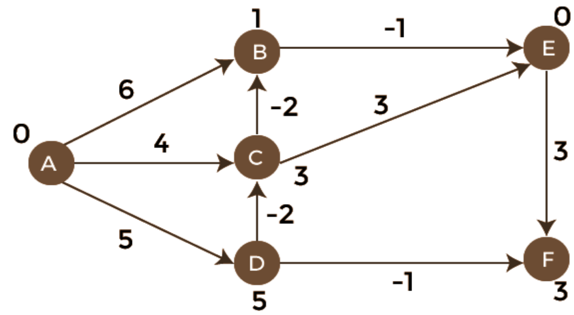


To find the shortest path of the above graph, the first step is note down all the edges which are given below: (A, B), (A, C), (A, D), (B, E), (C, E), (D, C), (D, F), (E, F), (C, B) .Let's consider the source vertex as 'A'; therefore, the distance value at vertex A is 0 and the distance value at all the other vertices as infinity shown as below:

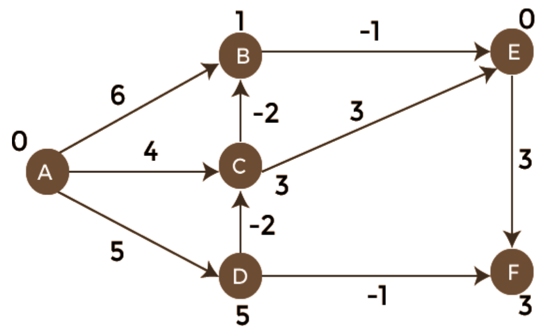
First Iteration:



Second iteration:



Third iteration:



Here there is no updation* after third iteration.

We again check all the edges. The first edge is (A, B). Since (0 + 6) is greater than 1 so there would be no updation in the vertex B.The next edge is (A, C). Since (0 + 4) is greater than 3 so there would be no updation in the vertex C.The next edge is (A, D). Since (0 + 5) equals to 5 so there would be no updation in the vertex D.The next edge is (B, E). Since (1 - 1) equals to 0 which is less than 5 so update:

d(v) = d(u) + c(u, v) , d(E) = d(B) +c(B , E) = 1 - 1 = 0.

The next edge is (C, E). Since (3 + 3) equals to 6 which is greater than 5 so there would be no updation in the vertex E.The next edge is (D, C). Since (5 - 2) equals to 3 so there would be no updation in the vertex C.The next edge is (D, F). Since (5 - 1) equals to 4 so there would be no updation in the vertex F.

The next edge is (E, F). Since (5 + 3) equals to 8 which is greater than 4 so there would be no updation in the vertex F.The next edge is (C, B). Since (3 - 2) equals to 1` so there would be no updation in the vertex B.

Improved Average Subjective Bell Man Ford Procedure (IASBMFP):

Procedure:

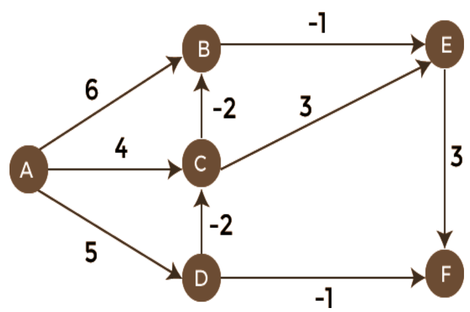
Step 1: Assign weight of starting node(vertex) is zero.

Step 2: Consider linked node from starting node(in all ways) and calculate degree of the corresponding node D and add all weight of the edges W(E)and weight of the corresponding node store in W(node).

Step 3: Calculate Average weight of the linked node is AW(node) = (W(E)+W(node)) / (D+1).

Step 4: For every node, repeat steps 2 & 3 upto there is no updation.

Improved Average Subjective Bell Man Ford Procedure:



Weigt of the Nodes, after applying IASBMFP is A = 3.75, B = 1, C= 1.2, D= 1.75, E= 1.25, F= 1.6667.

Weigt of the Nodes, before applying IASBMFP that is , by BMFP is A = 0, B = 1, C= 3, D= 5, E= 0, F=3.

Nature of the Weigt of the Nodes with BMFP and IASBFP:

Name of the Node	BMFP	IASBMFP	Compartion
A	**0**	**3.75**	**Increased**
B	**1**	**1**	**No change**
C	**3**	**1.2**	**Decreased**
D	**5**	**1.75**	**Decreased**
E	**0**	**1.25**	**Increased**
F	**3**	**1.6667**	**Decreased**
Total Among all Nodes	**12**	**10.617**	**Decreased**
Over all performance after applying IASBFP is decreased (in %) .			**11.10833%**

Conclusion:

The proposed Procedure

i. Provides very effective in network transportation traffic system.

ii. Will be very useful for network capacity determination with perfection.

iii. Adopting optimum in cost manner.

Acknowledgement:

The Authors would like to thank Prof. Ponnammal Natarajan, Former Director-Research, Anna University-Chennai. also the authors would like to extend sincere thank to Prof. K Sarukesi, Former Vice-Chancellor, Hindustan Institute Of Science And Technology-Chennai.

REFERENCES

[1] Gallian J.A., & quot; A Dynamic Survey of Graph Labeling ",The Electronic journal of Combinatorics, 2015.

[2] J.A. Bondy and U.S.R. Murty, "Graph Theory with Applications", London, Macmillan 1976.

[3] Harrary, F. “Graph theory”, Addison Wesley, Reading Massachusetts, USA, 1969.

[4] M. Chelali, L.Volkmann, “Relation Between the Lower Domination parameters and the Chromatic number of a Graph”, Discrete Mathematics 274, 2004, 1-8.

[5] K. Thiagarajan and P. Mansoor, “Expansion of Network Through Seminode”, IOSRD International Journal of Network Science,Vol 1, Issue 1, 2017,7-11.