**MINOR PROJECT 1**

**SYNOPSIS**

on

**Comparative Analysis of shortest path algorithms on the basis of graphical plots**

**Submitted By**

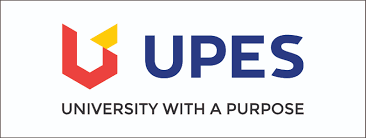
|  |  |  |  |
| --- | --- | --- | --- |
| Siddhi Gupta(61) | Yogesh(69) | Tushar Goyal(72) | Bhavuk Baluja(73) |
| 500067967 | 500069549 | 500068373 | 500070089 |
|  |  |  |  |

**Under the Guidance of**

**Ms. Kalpana Rangra**

Assistant Professor

Department of Cybernetics



School of Computer Science

University of Petroleum and Energy Studies

Dehradun - 248007

**Table of Content**

**Contents:**

1. Abstract 1
2. Introduction 1
3. Problem Statement 2
4. Literature 2
5. Objectives 5
6. Methodology 5
7. System Requirements(Software/Hardware) 6
8. Schedule (Pert Chart) 6
9. References 7

**List of Figures**

**List of Figures**

1. Pert Chart 6

**Synopsis Report**

**Project Title:-**

Comparative Analysis of shortest path algorithms on the basis of

graphical plots

# **1. Abstract:**

The main reason to choose this topic as a Minor Project is to compare and analyse the algorithms for finding the Shortest Path by using the Dijkstra’s, Bellman-Ford and Floyd-Warshall AlgorithmComparing, analyzing and plotting the graphs of each algorithm makes it easy to understand which algorithm is best for finding the shortest path for a particular type of question.

This project is created to find the shortest path very easily. Shortest Path Algorithm states that the algorithm to find the path which has minimal Distance(or path) between two nodes(or vertices). There are some Examples where the shortest path algorithm can be used like- telecommunications, network routing protocols, route planning, traﬃc control, path ﬁnding in social networks, computer games, and transportation systems, etc[1].

.

**Keywords:**  Dijkstra’s Algorithm, Bellman-Ford Algorithm, Floyd-Warshall Algorithm.

# **2. Introduction:**

Today's requirement for the machine is less time to complete the work. Finding the shortest path or route from a source to a destination is important. The Shortest path problem is to find a path between two vertices or nodes in a graph such that the sum of the weights of its edges is minimized. There are some examples of using shortest path algorithms such as web mapping like: Google Maps, Transportation Systems, Computer Networks like the Internet etc. [1]The very common example of finding the shortest path is Rubik's cube, there are vertices or nodes to represent the states of puzzle and edges correspond to a single turn by which a person can understand the minimum possible number of moves. In this project, the shortest path is illustrated with the help of graphs. A graph is a mathematical or pictorial representation which contains vertices and edges. With the help of edges it is possible to walk from one vertex to another vertex because edges are a particular type of line segment joining two vertices[2]. By drawing the graph of any problem then it will be much easier to understand, so it helps people to get knowledge about the concept of the question in a short period of time.

For Example:

* A graph could be made for a routing protocol in which nodes have information about the network which is directly attached to it and directed arcs help to pass information and data packets between the nodes.
* In road networks, we make a graph in which nodes represent the road junctions and edges or directed arcs help to connect the segment between two junctions.
* It is the topology communication where graphs are made in which nodes having switching elements and the directed arcs help to link between switching elements.

# **3. Problem Statement:**

There are several projects for finding the shortest path but they all give the output in written form, which is difficult to understand and it takes more time to understand the shortest path.

Therefore, there is a need for a project which compares all the algorithms on the basis of their time complexity and gives the output in written form as well as in graphical form step by step which makes it easy to understand the shortest path.

**4. Literature:**

To understand the shortest path algorithm we need to know in detail about Graph, Dijkstra’s Algorithm, Bellman-Ford Algorithm and Floyd-Warshall Algorithm.

1. **Graph representation and Explanation:** Graph is a mathematical representation that represents data in an organized manner. Graphs contain edges(E) and vertices(V). Edges are the lines that connect two vertices and vertices are the points or the nodes. The shortest path problem is represented by the graph whether the graph is directed, undirected or mixed[2].
2. **Dijkstra’s Algorithm representation and explanation:** Dijkstra’s Algorithm is a single-source shortest path problem algorithm which has non-negative edges weight. Single source shortest path states that finding the shortest path from the source vertex to all other vertices[3].The node at which we are starting is called the initial node. It was discovered by the computer scientist **Edsger W. Dijkstra** in 1956.

Time complexity: O(E Log V)

1. Best Case Complexity = O(E+V log V)
2. Average Case Complexity = O(E+V log V)
3. Worst Case Complexity = O(E+V log V)
4. Time complexity is O(E+V^2), if the priority queue is not used.

Space complexity = O(V)

where, E is the number of edges and V is the number of vertices.

Dijkstra’s algorithm is mainly used to find the shortest path in social networking, telephone network, finding the best route in the map etc.

There is several step to work on algorithm of Dijkstra's algorithm:

**Step 1-** Take the weighted graph.

**Step 2-** Take a starting vertex and assign value zero to it and assign all other vertices values infinite.

**Step 3-** Go to each vertex and update its path and distance.

**Step 4-** If the distance of adjacent is not more than the new path distance then, don’t update.

**Step 5-** Not visit the nodes which are already updated in the above steps.

**Step 6-** After visiting each vertex make the graph which contains the shortest path[4].

1. **Bellman-Ford Algorithm representation and explanation:** Bellman-Ford Algorithm is a single-source shortest path problem algorithm which can have negative edge weight also. It was founded by Alfonso Shimbel in 1955, yet is rather named after Richard Bellman and Lester Ford Jr., who distributed it in 1958 and 1956. Bellman-Ford algorithm works on the path from the starting vertex to the other vertices. It is more versatile and has the capability to solve the problem by the graph which contains negative edge weights. In real life, some examples where negative weight edges can exist like cash flow, heat released in a chemical reaction, heat absorbed in chemical reaction etc[3]. If a graph contains a "negative cycle" (a cycle whose edges sum is negative value).

Time complexity :

1. Best Case Complexity = O(E)
2. Average Case Complexity = O(VE)
3. Worst Case Complexity = O(VE)

Space complexity = O(V)

where, E is the number of edges and V is the number of vertices.

Bellman-Ford algorithm is mainly used to find the shortest path, calculating shortest paths in routing etc.

There is several step to work on algorithm of Bellman-Ford:

**Step 1-** Take the weighted graph.

**Step 2-** Take a starting vertex and assign value to it zero and assign all other vertices values infinite.

**Step 3-** Use this formula (for each edge A-B)

If Distance[B]>Distance[A] + weight of edge A and B

then update Distance[B]

Distance[B] = Distance[A] + weight of edge A and B

**Step 4-** We can do step 3 for (V-1) times and V is the number of vertices in the graph.

**Step 5-** It tells if there is a negative weight cycle in the graph. Check for each edge A-B:

If Distance[B]>Distance[A] + weight of edge A and B

**Step6-** After visiting each vertex make the graph which contains the shortest path.[5]

1. **Floyd-Warshall Algorithm representation and explanation:** Floyd-Warshall Algorithm is used for finding the shortest path between all pairs of vertices. The Floyd-Warshall Algorithm was proposed by **Robert Floyd** in 1962. The graph used in Floyd-Warshall Algorithm is a weighted graph with positive or negative edge weights. The Floyd-Warshall Algorithm is used for both directed and undirected weighted graphs. It is applied in negative edges but not for negative cycles(sum of edges in cycle is equal to negative value)[3]. Floyd-Warshall Algorithm is also known by different names as Floyd's algorithm, Roy-Floyd algorithm, WFI algorithm or Roy-Warshall algorithm.

Time complexity: O(V3)

1.Best Case Complexity =O(V3)

2.Average Case Complexity = O(V3)

3.Worst Case Complexity = O(V3)

Space complexity = O(V2)

where, V is the number of vertices.

Floyd-Warshall Algorithm is mainly used to find the shortest path in directed graphs, find the Inversion of real matrices, find the transitive closure of directed graphs, etc.

There are several step to work on algorithm of Bellman-Ford:

**Step 1-** Create a matrix of dimension (w\*w) but here we take (4\*4) matrix. where,w is the number of vertices. The row and also the column are unit indexed as I and j resp. where,i and j are unit vertices of the graph.

Make a matrix and fill the cell in the matrix with the distance between ith and jth

vertex.

**Step 2-** Create a matrix B1 using matrix B0 and fill the element in matrix B1 by help of formula but the elements in the first column and the first row are left as they are

B[i][j] =(B[i][k] + B[k][j]) if (B[i][j] > B[i][k] + B[k][j])

where, k is the vertex=1.

**Step 3-** B2 is created using B3. The elements in the second column and the second row are left as they are. Then, repeat the step of 2 by putting k vertex of value is 2.

**Step 4-** Same way B3 and B4 is also created.

**Step 5-** B4 gives the shortest path.

**Step 6-** Make a graph of matrix B4[6].

# **5. Objective:**

The main objective of this project is to compare all the algorithms by plotting their graphs on the basis of their time complexity and give the output in written form as well as in graphical form step by step which makes it easy to understand the shortest path and tells which algorithm should be used to solve the question more efficiently for a particular type of question.

# **6. Methodology:**

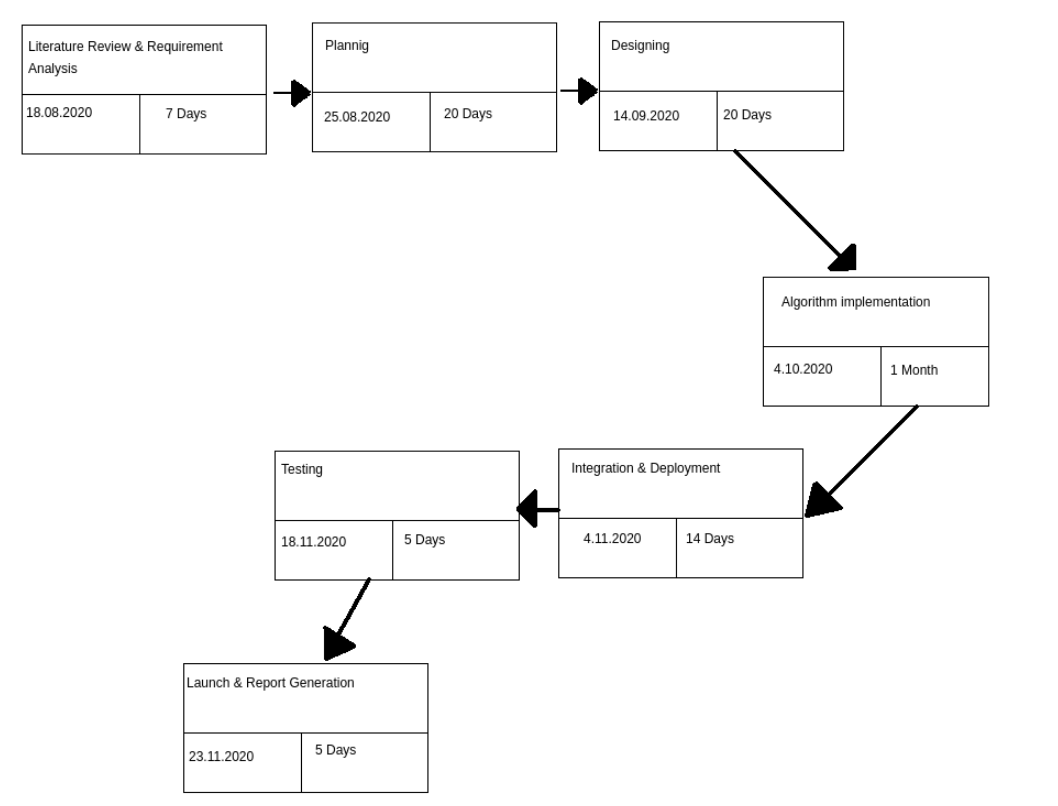
# In our project, we are comparing, analysing and plotting the graphs of each algorithm which algorithm is best for finding the shortest path for a particular type of question.

* Design and development of prerequisites:
* Preparing the data flow diagram of the project.
* Implementation, Comparison and Analysis of Algorithms:
* Implementing different algorithms.
* Comparing the algorithms by plotting their graphs on the basis of their time complexity.
* Plotting the graphs of shortest path step by step using different algorithms.
* Integrating the modules.
* Testing:
* Testing the integration of modules.
* Testing the system.

# **7. System Requirement:**

* Operating System-
* Linux/ Windows/ MacOS or any other OS.
* Software-
* gcc compiler or any other online ‘C’ Compiler like: GDB compiler.
* Hardware-
* Memory - 2GB (Recommended)
* Storage - 4GB (Recommended)

**8. PERT Chart:**



**Figure: 1**

# **9. References:**

1. Amgad Madkour, Walid G. Aref, Faizan Ur Rehman, Mohamed Abdur Rahman, Saleh Basalamah. *A Survey of Shortest-Path Algorithms*. Purdue University, West Lafayette, USA, Umm Al-Qura University, Makkah, KSA, may 8, 2017.
2. Kairanbay Magzhan, Hajar Mat Jani ― *A Review And Evaluations Of Shortest Path Algorithms*, the international journal of scientific & technology research volume 2, issue 6, june 2013, pp. 99-104.

# ‘Shortest path problem’(8 September 2020)*Wikipedia.*Available at: https://en.wikipedia.org/wiki/Shortest\_path\_problem(Accessed: 25 August 2020).

# ‘Dijkstra’s shortest path algorithm’(22 April 2020)*GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/(Accessed: 26 August 2020).

# ‘Bellman–Ford Algorithm’(23 April 2020)*GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/(Accessed: 28 August 2020).

# ‘Floyd Warshall Algorithm’(18 July 2019)*GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/floyd-warshall-algorithm-dp-16/(Accessed: 28 August 2020).

**Approved By**

**(Ms. Kalpana Rangra) (Dr. Monit Kapoor)**

**Project Guide Head of Department**