**Shortest Path Algorithm**

### Project-I

**BACHELOR OFTECHNOLOGY**

(Artificial Intelligence and Data Science.)



## SUBMITTEDBY:

## Name- Alis Thakur , Alok Kumar ,

## Amandeep , Ananya

## Amit Verma

## Roll no.- (2330684), (2330685)

## (2330686), (2330687),

## (2330688)

## Date – 14/02/2025

**UndertheGuidanceof**

Name of Mentor- Mr. Mayank

Designation of mentor

# DepartmentofArtificial Intelligence and Data Science

# Chandigarh Engineering College Jhanjeri Mohali - 1040307

## TableofContents

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Contents** | **Page No** |
| 1. | Introduction | 1 |
| 2. | BriefLiteraturesurvey | 2 |
| 3. | Problemformulation | 3 |
| 4. | Objectives | 3 |
| 5. | Methodology/ Planningofwork | 4 |
| 6. | Facilitiesrequiredforproposedwork | 4 |
| 7. | References | 5 |

1. **Introduction**

The shortest path algorithms is a fundamental issue in graph theory and computer science, with applications in areas such as routing, navigation, transportation, and network optimization. Several algorithms have been developed to find the shortest path between two nodes in a graph, each with different time and space complexities, as well as specific strengths and weaknesses. This research aims to conduct a comparative study of various shortest path algorithm, focusing on their performance, efficiency, and suitability for different types of graphs and real world scenarios.

This study provides a comparative analysis of various shortest path algorithms, including Dijkstra’s Algorithm, Bellman-Ford Algorithm,Floyd-Warshall Algorithm, and Johnson’s Algorithm.

## Abstract-

## The shortest path problem is a fundamental challenge in graph theory and has

## widespread applications in network routing, transportation, robotics, and

## geographical mapping. This study provides a comparative analysis of various

## shortest path algorithms.

## 1.2 Introduction to Shortest Path Problem-

The shortest path problem is a well-known problem in graph theory and computer science.

It seeks to determine the path between two nodes in a graph such that the total weight of

the path isminimized. This problem has applications in fields such as transportation

networks, computer networks, robotics, and geographic information systems (GIS).

**2. Brief Literature Survey**

1. **Dijkstra's Algorithm (1959)**: One of the earliest and most widely used algorithms for solving the single-source shortest path problem is Dijkstra's Algorithm. Proposed by Edsger Dijkstra, it efficiently finds the shortest path from a source node to all other nodes in a graph with non-negative edge weights. The algorithm uses a greedy approach and priority queues for optimal performance.
2. **Bellman-Ford Algorithm (1958)**: The Bellman-Fordalgorithm is another significant contribution that can handle graphs with negative weight edges. Although it is slower than Dijkstra’s algorithm, it is capable of detecting negative-weight cycles, making it versatile for a wider range of applications.
3. **A Algorithm (1968)**:The A algorithm is a popular extension of Dijkstra's algorithm, which incorporates heuristics to improve performance in certain types of graphs, especially in pathfinding problems such as navigation. By prioritizing paths that seem closer to the goal, it reduces the number of nodes that need to be explored.
4. **Floyd-Warshall Algorithm (1962)**: The Floyd-Warshall algorithm is a dynamic programming approach for finding the shortest paths between all pairs of nodes in a graph. It is known for its simplicity and effectiveness on smaller graphs, although it has a higher time complexity compared to other algorithms like Dijkstra.
5. **Johnson's Algorithm (1977)**: Johnson’s Algorithm is a more recent development that combines the Bellman-Ford and Dijkstra algorithms to solve the all-pairs shortest path problem efficiently. It works well for sparse graphs and is often preferred in cases where multiple queries are made regarding the shortest path.
6. **Optimizations and Variations**: In recent years, various modifications and optimizations of these classical algorithms have been proposed. These include parallel versions, implementations for real-time systems, and algorithms designed for specific graph structures like grid graphs, road networks, or social networks.

# Problem Formulation

**3.1 Needs of algorithms**-

1. **Navigation and Route Planning**: In transportation systems (like roads, railways, or flight networks), determining the fastest or most cost-effective route between two points isessential. Applications like GPS systems and delivery route optimization rely on shortestpath algorithms.
2. **Network Optimization**: In communication and computer networks, shortest path algorithmsare used to find the quickest path for data packets to travel across the network, improving speed and efficiency. This is important for minimizing latency and ensuring reliable data transmission.
3. **Resource Allocation**: In logistics, project management, and various scheduling problems, finding the shortest or least costly path helps in resource allocation, reducing waste, and optimizing workflows.
4. **Robotics and AI**: Autonomous robots and AI systems use shortest path algorithms to navigate their environment efficiently, avoiding obstacles and finding the optimal route for tasks like delivery, exploration, or pathfinding in virtual environments (e.g., video games).
   1. **Objectives**

**4.1**In the context of an algorithm, such as the shortest path algorithm, the objectives couldinclude:

1. **Minimizing Distance:** The goal is to find the shortest possible path between two points in a network (such as the shortest road route between two cities).
2. **MinimizingCost:** The objective could be to minimize the financial or resource cost of transversing a network, like minimizing fuel usage in a transportation network.
3. **Minimizing Time:** In some cases, the goal is to reduce the time it takes to travel from one point to another, such as finding the quickest path in a delivery system.
4. **Optimizing Efficiency:** The objective might focus on optimizing the overall system performance by reducing bottlenecks or increasing throughput in network systems or processes.
   1. **Methodology**

**5.1 This study compares algorithms based on:**

**1. Problem Definition**

* Objective: Clearly define the goal (e.g., find the shortest path in a network, minimize cost, or time).
* Input/Output: Understand what the inputs (e.g., graph, network, nodes, edges) and outputs (e.g., shortest path, total distance) will be.

**2. Selection of Algorithm**

* Choose the appropriate algorithm based on the problem’s characteristics:
* Dijkstra’s Algorithm: Common for graphs with non-negative weights.
* Bellman-Ford Algorithm: Works with graphs that have negative edge weights.
* A Algorithm: Used when additional heuristics can speed up the search.

**3. Graph Representation**

* Represent the problem as a graph with nodes (representing locations) and edges (representing connections between locations with costs or distances).
* Determine whether the graph is directed or undirected, weighted or unweighted.

**4. Algorithm Execution**

* Implement the selected algorithm and run it on the input data. This will typically involve:
* Initializing data structures (like priority queues for Dijkstra’s algorithm).
* Iteratively exploring the graph, updating distances, and tracking paths.

**5. Result Evaluation**

* Assess the outcome (e.g., shortest path found).
* Ensure the solution satisfies the problem’s objective (minimizing distance, time, or cost).
* Perform validation and testing using different cases to ensure accuracy.

**5.2Facilities required for proposed work-**

* Hardware: Computers
* Learning Resources: Research papers, tutorials, and textbooks

**6. References**

* **Research papers-**
* "A Note on Two Problems in Connexion with Graphs" by E. W. Dijkstra: (1959): The original paper describing Dijkstra's algorithm.
* "On a Routing Problem" by R. Bellman: (1958): A foundational paper on shortest path problems, including the Bellman-Ford algorithm.
* "Algorithm 97: Shortest Path" by R. W. Floyd: (1962): The paper introducing the Floyd-Warshall algorithm.
* **Online Sources-**
* Wikipedia