**Date:22.10.25**

**TASK:11**

# Implementation of Vehicle Routing Problem

**CO1, CO2, CO3 S3**

# PROBLEM STATEMENT

The Vehicle Routing Problem (VRP) aims to determine the most efficient routes for a fleet of vehicles to deliver goods or services to a set of customers. Each vehicle starts and ends at a depot, and every customer must be visited exactly once. The goal is to minimize total travel distance or cost while satisfying vehicle capacity and route constraints.

# AIM

To implement and analyze a Vehicle Routing Problem (VRP) solution using AI optimization techniques for minimizing the total travel distance and cost.

# OBJECTIVE

* To understand the working of routing optimization problems.
* To implement an algorithm that minimizes the total route cost or distance.
* To ensure all customers are served without exceeding vehicle capacity limits.
* To apply heuristic or AI-based methods for solving real-world logistics problems.

# DESCRIPTION

The Vehicle Routing Problem is a generalization of the Traveling Salesman Problem (TSP). In VRP, multiple vehicles are used instead of one, and each vehicle must service a subset of customers starting and ending at a depot. Since VRP is NP-hard, exact solutions become computationally expensive for larger instances. Therefore, heuristic or AI-based algorithms such as Genetic Algorithms, Ant Colony Optimization, or Greedy Search are commonly used to obtain near-optimal solutions efficiently.

# ALGORITHM

1. Start from the depot as the initial location.
2. Select the nearest unvisited customer based on the minimum distance.
3. Move to that customer and mark it as visited.
4. If vehicle capacity is full or all customers are served, return to the depot.
5. Repeat the process for remaining customers and vehicles.
6. Calculate the total distance covered by all routes.
7. Display the optimal or near-optimal route and total cost.

# PROGRAM

import numpy as np dist\_matrix = [

[0, 10, 15, 20],

[10, 0, 35, 25],

[15, 35, 0, 30],

[20, 25, 30, 0]

]

num\_locations = len(dist\_matrix) visited = [False] \* num\_locations visited[0] = True # start at depot route = [0]

total\_distance = 0

current\_location = 0

for \_ in range(num\_locations - 1): next\_location = None min\_distance = float('inf')

for j in range(num\_locations):

if not visited[j] and dist\_matrix[current\_location][j] < min\_distance: min\_distance = dist\_matrix[current\_location][j]

next\_location = j route.append(next\_location) visited[next\_location] = True total\_distance += min\_distance

current\_location = next\_location

# Return to depot

total\_distance += dist\_matrix[current\_location][0] route.append(0)

print("Optimal Route (Approx):", route) print("Total Distance:", total\_distance)

# OUTPUT

****

**CONCLUSION**

The Vehicle Routing Problem was successfully implemented using a heuristic (nearest neighbor) approach. The algorithm efficiently determined an approximate optimal route with minimal travel distance. This implementation demonstrates how AI and optimization techniques can be applied to real-world logistics and transportation problems.