# TASK:5

Implementation of **Ant Colony Optimization** to Optimize Ride-Sharing Trip Duration using Python by following constraints.

**Aim:** To Implement Ant Colony Optimization to Optimize Ride-Sharing Trip Duration using Python.

**Algorithm:**

**Step 1:**[Initialization]

t=0;NC=0;for each edge (I,j),initialize trail intensity.

**Step 2:**[starting node]

For each ant k:place ant k on a randomly chosen city and store this information in tablet.

**Step 3:**Build a tour for each ant.

**Step 4:**global update of trail.

**Step 5:** termination conditions,memorize the shortest tour found to this point.

**Program:**

import numpy as np

from numpy import inf

d = np.array([

[0, 10, 12, 11, 14],

[10, 0, 13, 15, 8],

[12, 13, 0, 9, 14],

[11, 15, 9, 0, 16],

[14, 8, 14, 16, 0]

])

iteration = 100

n\_ants = 5

n\_citys = 5

m = n\_ants

n = n\_citys

e = 0.5

alpha = 1

beta = 2

visibility = 1 / d

visibility[visibility == inf] = 0

pheromne = 0.1 \* np.ones((n, n))

rute = np.ones((m, n + 1))

for ite in range(iteration):

rute[:, 0] = 1

for i in range(m):

temp\_visibility = np.array(visibility)

for j in range(n - 1):

combine\_feature = np.zeros(n)

cum\_prob = np.zeros(n)

cur\_loc = int(rute[i, j] - 1)

temp\_visibility[:, cur\_loc] = 0

p\_feature = np.power(pheromne[cur\_loc, :], alpha)

v\_feature = np.power(temp\_visibility[cur\_loc, :], beta)

combine\_feature = np.multiply(p\_feature, v\_feature)

total = np.sum(combine\_feature)

probs = combine\_feature / total

cum\_prob = np.cumsum(probs)

r = np.random.random\_sample()

city = np.nonzero(cum\_prob > r)[0][0] + 1

rute[i, j + 1] = city

left = list(set([i for i in range(1, n + 1)]) - set(rute[i, :-2]))[0]

rute[i, -2] = left

rute\_opt = np.array(rute)

dist\_cost = np.zeros((m, 1))

for i in range(m):

s = 0

for j in range(n - 1):

s += d[int(rute\_opt[i, j]) - 1, int(rute\_opt[i, j + 1]) - 1]

dist\_cost[i] = s

dist\_min\_loc = np.argmin(dist\_cost)

dist\_min\_cost = dist\_cost[dist\_min\_loc]

best\_route = rute[dist\_min\_loc, :]

pheromne = (1 - e) \* pheromne

for i in range(m):

for j in range(n - 1):

dt = 1 / dist\_cost[i]

pheromne[int(rute\_opt[i, j]) - 1, int(rute\_opt[i, j + 1]) - 1] += dt

print("route of all the ants at the end :")

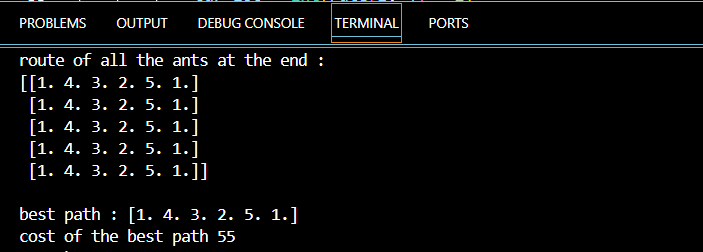
print(rute\_opt)

print()

print("best path :", best\_route)

print("cost of the best path", int(dist\_min\_cost[0]) + d[int(best\_route[-2]) - 1, 0])

**Output:**



**Result:**

Thus the Implementation of Ant Colony Optimization to Optimize Ride-Sharing Trip Duration using Python was successfully executed and output was verified.