CSCE 5210 SECTION 002
FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE
(FALL 2022 1)

PROJECT-1 MILESTONE 1

MILESTONE 1 R1:

Pseudo Code to adapt the shortest path between a pair of nodes:

This Pseudo Code is based on the DIJKSTRA ALGORITHM:

a) Select two nodes in the graph and label them S and D for the starting and Destination node respectively.

b) For each node in the graph set the optimal path as infinity or the largest possible number from the source initially and zero for the source itself. We initially set the source node as the current node.

c) Take a list which contains all the nodes that are unvisited.

d) Update the optimal path to all the nodes that are directly connected to the present node by checking, if the (current optimal path to the node) is more than (the optimal path to the current node plus the distance between the current node and selected node).

e) Remove the current node from the list of unvisited nodes.

f) Repeat the step (d) & (e) until all the nodes are visited.

g) The distances which are stored gives us the shortest path from the source node S to all the nodes in the graph. From them we select the distance between S & D.

MILESTONE 1 R2:

We need to include the following functionalities-

- Generate new navigation requests.

- Revise existing journeys.

- Keep track of journey time for each vehicle that has completed its journey.

- Keep a running count of total journey time taken across all vehicles that have completed their journeys.

Pseudo Code:

a) Using Random class, we generate 10 random pairs of nodes for every 1-Hour.

b) For every pair we calculate the shortest path to take using the current variables using the algorithm mentioned in Milestone 1 R1 as follows: -

1) Select the pair for which we need to calculate the shortest path and label them S and D for the starting and Destination node respectively.
2) For each node in the graph set the optimal path as infinity or the largest possible number from the source initially and zero for the source itself. We initially set the source node as the current node.
3) Take a list which contains all the nodes that are unvisited.
4) Update the optimal path to all the nodes that are directly connected to the present node by checking, if the (current optimal path to the node) is more than (the optimal path to the current node plus the distance between the current node and selected node).
5) Remove the current node from the list of unvisited nodes.
6) Repeat the step (4) & (5) until all the nodes are visited.
7) The distances which are stored gives us the shortest path from the source node S to all the nodes in the graph. From them we select the distance between S & D.

c) For every clock tick we update the base traffic load for all edges in the network according to value of p from the normal distribution.

d) After updating the edges we once more calculate the time that will take if we follow the current path using the newly updated variables.

e) If the time required does not increase by 5% or more we use the same path, else we calculate a new shortest path. This new calculation will not be made from the beginning of the journey but from the current node where the car is at as follows: -

1) Get the node where the car is currently at label it S for starting and D for Destination node.
2) For each node in the graph set the optimal path as infinity or the largest possible number from the source initially and zero for the source itself. We initially set the source node as the current node.
3) Take a list which contains all the nodes that are unvisited.

4) Update the optimal path to all the nodes that are directly connected to the present node by checking, if the (current optimal path to the node) is more than (the optimal path to the current node plus the distance between the current node and selected node).
5) Remove the current node from the list of unvisited nodes.
6) Repeat the step (4) & (5) until all the nodes are visited.
7) The distances which are stored gives us the shortest path from the source node S to all the nodes in the graph. From them we select the distance between S & D.

f) Repeat steps (c), (d) & (e) for every clock tick i.e. for every 15 Minutes.

g) Repeat steps (a) & (b) for every 4 clock ticks i.e. for every 1 Hour.

Note: Variables mentioned in steps (b) & (d) are the base traffic loads, value of p.