

NoDelayWorry

Team Name: TheStarks

Team Members:

1. Soumya Taurani
2. Parul Shandilya
3. Pooja Srinivas
4. Komal Kungwani

Problem statement :

We aim to automate the process of delivery routing of trucks such that, the fuel cost is minimized and the customer satisfaction is maximized. Customer Satisfaction is achieved by ensuring the availability of good quality(fresh) inventory at shops. This is achieved by ensuring the trucks reach the shops without delay and reach the goal with minimum fuel costs.

Algorithm:

1. Reinforcement learning is a type of Machine Learning where an agent learns how to behave in an environment by performing actions and seeing the results.
2. In our case, the agents will be delivery trucks and these agents will receive rewards based on fuel consumption and customer misses.
3. Q-learning Algorithm is used to provide a best route with the help of Customer arrival rate, Shop location, warehouse location, Maximum Truck load..

Tools used:

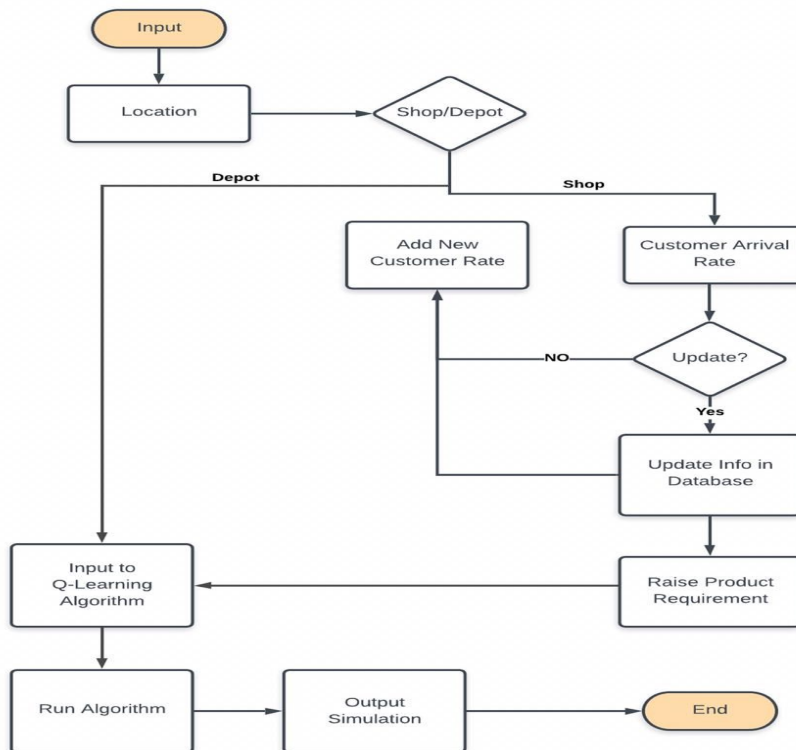
1. Backend: Python, Java
2. Front End: HTML, CSS, JQuery
3. Version Control — Git command line.
4. Text Editor — Eclipse Oxygen.
5. Web Browsers — Google Chrome + Mozilla Firefox at least.
6. Browser developer tools — Chrome/Firefox developer tools.

Technical Architecture:

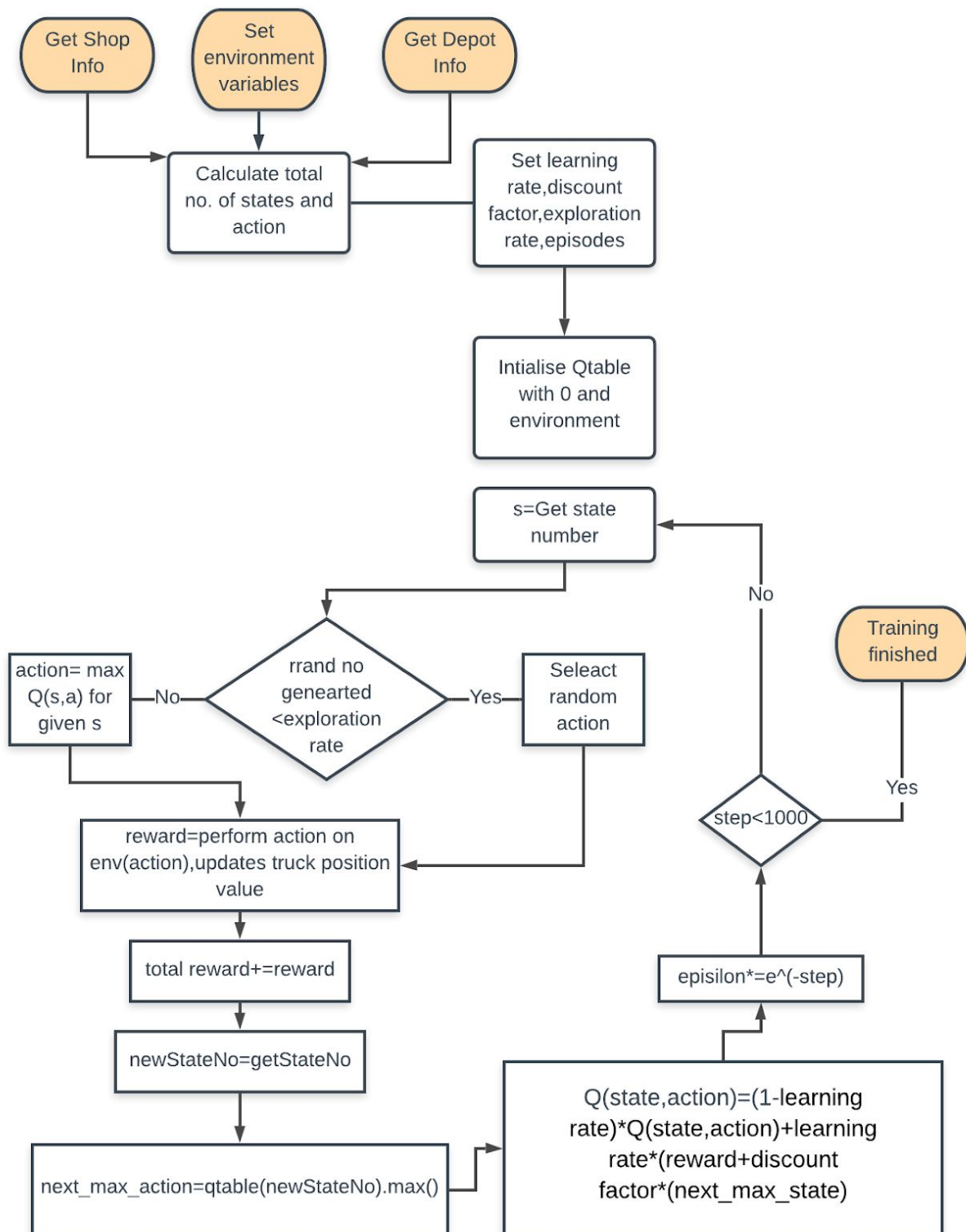
1. Vehicle Routing: Vehicle routing consists of 3 points.
 - Truck load: Trucks replenish their inventory from a depot. They subsequently deliver their load to the concerned shops. Truck inventories are discretized into various levels.
 - Fuel costs: These are represented by a small negative reward for every “move” action. Waiting is free.
 - Shop priority: The priority of a shop is decided by its current stock. In case a customer arrives and finds the stock empty, a large negative reward is added.
2. Inventory Control: Inventory control depends on 3 points, namely:
 - Purchases: Customer consumption is modeled by assigning some probability with which the shop’s inventory is decreased by 1 unit.

- Replenishments: A shop's inventory is increased by a certain amount when a delivery truck arrives at it.
- Shop priority: The priority of a shop is decided by its current stock. In case a customer arrives and finds the stock empty, a large negative reward is added.

Flow Diagram:



```
Activities Terminal Sun 03:39 ● lenovo@lenovo-Lenovo-G50-80: ~/Downloads/rlproductdelivery-master
File Edit View Search Terminal Help
Episode: 0
new currCord : 47
new currCord : 37
new currCord : 42
new currCord : 56
new currCord : 34
new currCord : 46
Episode: 1000
new currCord : 22
new currCord : 16
new currCord : 52
new currCord : 53
new currCord : 4
new currCord : 19
Episode: 2000
new currCord : 3
new currCord : 32
new currCord : 28
new currCord : 2
new currCord : 0
new currCord : 10
Episode: 3000
new currCord : 20
new currCord : 18
new currCord : 23
new currCord : 8
new currCord : 5
new currCord : 9
Episode: 4000
new currCord : 29
new currCord : 12
new currCord : 35
new currCord : 11
new currCord : 27
new currCord : 21
Episode: 5000
new currCord : 43
new currCord : 58
```



Input:

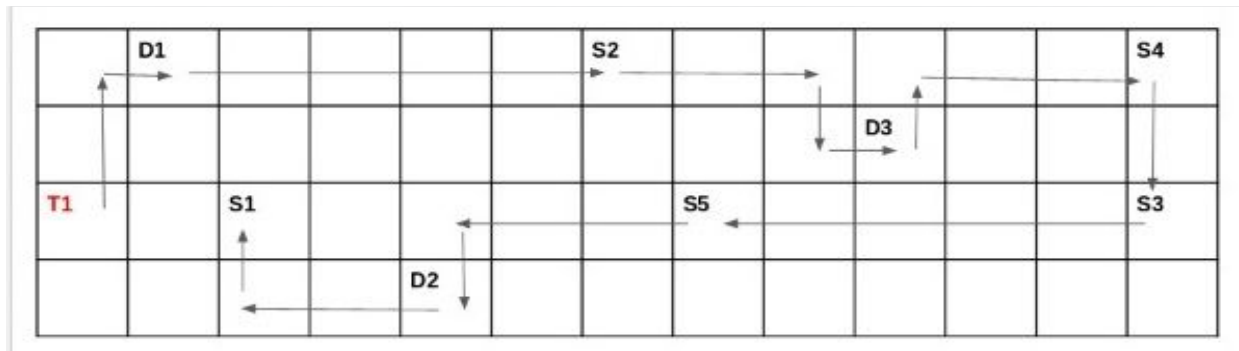
1. This is the Layout of a city describing the positions of shops and depots in the form of a grid.

2. Shops are represented by S1, S2, S3 in the grid.
3. Depos are represented by D1, D2, D3 in the grid.
4. Trucks are represented by T1, T2, T3 in the grid.

	D1					S2						S4
									D3			
T1		S1					S5					S3
				D2								

Output:

1. Optimal path of the truck.



Assumptions:

1. Real world is visualised as a grid.

Future Scope:

1. This can be extended to real world map instead of grid with a lot of rewards.
2. Apply deep Q learning in order to apply for greater no. of states.
3. Apply it to multi agent i.e. multi trucks along with communication.