SCT Hackathon Round 1

Route Optimisation

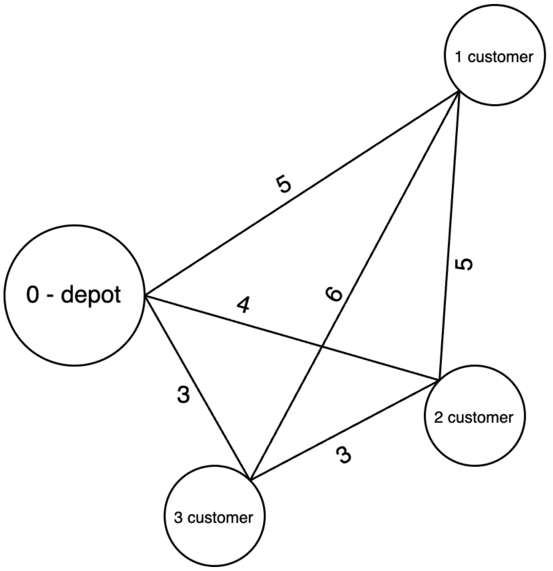
part A: Find the Best Delivery Route

Let’s say we have a depot/hub of an eCommerce organisation (like Walmart) and packages (customer orders) must be delivered from the depot/hub to customer locations.

Delivery Process Rules:

1. Vehicle will start from depot.
2. It will drive to every customer location who placed the order, deliver the package.
3. Vehicle will return back to the depot from the last customer delivery location.
4. Vehicle will visit each customer location only once.

Example delivery process/network represented as a Graph with its description



In the above figure

Nodes:

0: depot/hub node

1,2,3: customer nodes

Edges:

Number on each edge represents the traveling distance between the two nodes of the edge in kms.

1. distance from 0 (depot) to 1 (customer) is 5 kms.
2. distance from 0 (depot) to 2 (customer) is 4 kms.
3. distance from 0 (depot) to 3 (customer) is 3 kms.
4. distance from 1 (customer) to 2 (customer) is 5 kms.
5. distance from 1 (customer) to 3 (customer) is 6 kms.
6. distance from 2 (depot) to 3 (customer) is 3 kms.

Possible delivery routes that vehicle can take:

Vehicle always starts at 0 (depot/hub) and ends at 0 (depot/hub)

1. **Route 1** : Vehicle starts at 0 , then visits 1, then 2, then 3 and return back to 0. Distance travelled if vehicles takes this route (5 (0- >1) + 5(1- >2) + 3(2- >3) + 3(3- >0)) is 16 kms
2. **Route 2** : Vehicle starts at 0 , then visits 1, then 3, then 2 and return back to 0. Distance travelled if vehicles takes this route (5 (0- >1) + 6(1- >3) + 3(3- >2) + 4(2- >0)) is 18 kms

Route 1 is better than route 2 in this case since vehicle will travel less distance if it takes Route 1.

In this example there are two possible routes which minimise the distance travelled by vehicle the most, delivering all the orders and returning back to the depot.

Route 1:

1. 0 - > 1 - > 2 - > 3 - > 0 (Route)
2. 16 kms (Distance travelled)

Route 2:

1. 0 - > 3 - > 2 - > 1 - > 0 (Route)
2. 16 kms (Distance travelled)

If Vehicle takes any other route it would have to travel more than 16 kms.

Problem statement:

Find the best delivery route you think vehicle should take

- which minimises the total distance travelled by the vehicle
- delivers all the customer orders following the rules of the delivery process mentioned above
- **If there are multiple best delivery routes possible , output any one of the those routes.**
- Traveling distance from a- >b is same as the distance from b- >a (a,b are nodes in the network where a can be depot/customer and b can be depot/customer)
- If vehicle wants to reach from node a to node b, it can do it only via direct edge (not via intermediate nodes) (a,b are nodes in the network where a can be depot/customer and b can be depot/customer)

Assume Vehicle has the capacity and time to deliver all the orders and there is only one vehicle for the delivery for this problem (part A)

Input and Output Dataset Format Details:

1. You would be given multiple input datasets.
2. Each input dataset would contain depot/hub details and customer order details.
3. Vehicle has to start from the depot/hub , deliver all the orders and return back to the depot **following the rules of the delivery process mentioned above**
4. **For each of the below input datasets**
    a. **Find the best delivery route you think vehicle should take**

Input Datasets:

5 datasets

1. part_a_input_dataset_1.csv (5 customer orders + 1 depot/hub)
2. part_a_input_dataset_2.csv (10 customer orders + 1 depot/hub)
3. part_a_input_dataset_3.csv (15 customer orders + 1 depot/hub)
4. part_a_input_dataset_4.csv (20 customer orders + 1 depot/hub)
5. part_a_input_dataset_5.csv (25 customer orders + 1 depot/hub)
    a. **If you face computational issues for this dataset, come up with a delivery route which minimises the distance travelled by vehicle the most.**

Metadata of input dataset:

Column Name	Column Description	Additional Details
order_id	Identifier for a customer order	
lng	delivery longitude of the customer	
lat	delivery latitude of the customer	
depot_lat	depot/hub latitude	This will be same for all the orders in the dataset (all orders will be delivered from single depot /hub)

depot_lng	depot/hub longitude	This will be same for all the orders in the dataset (all orders will be delivered from single depot /hub)

How to calculate traveling distance between Nodes (Depot- > Customer / Customer - > Customer):

Assume traveling distance between two pairs of lat/longs is **haversine distance** rounded off to 2 decimals.

1. Python Implementation of **haversine distance**

```
import numpy as np
def haversine(lat1, lon1, lat2, lon2):
    R = 6371  # radius of Earth in kilometers
    phi1 = np.radians(lat1)
    phi2 = np.radians(lat2)
    delta_phi = np.radians(lat2 - lat1)
    delta_lambda = np.radians(lon2 - lon1)
    a = np.sin(delta_phi/2)**2 + np.cos(phi1) * np.cos(phi2) * np.sin(delta_lambda/2)**2
    res = R * (2 * np.arctan2(np.sqrt(a), np.sqrt(1 - a)))
    return np.round(res, 2)
```

2. **Based on your choice of programming language you can use external libraries or implement the above haversine function to calculate distance between two pairs of lat/longs**
3. For example , If you are using python you can use library to calculate distance between two lat/longs

```
geopy.distance.geodesic
```

4. Example : Traveling distance between 43.8121,126.5669 and 43.81811,126.55716 is 1.03 kms (rounded off to 2 decimals using haversine /geodesic)

Output Datasets:

Example Input would look like:

order_id	lng	lat	depot_lat	depot_lng
123	126.59037	43.85294	43.871	126.5661
124	126.58835	43.85041	43.871	126.5661

Lets say your best path is : depot - > 123 - > 124 - > depot.

order number 123 is delivered first in this route - dlvr_seq_num is 1 (dlvr_seq_num - Delivery Sequence Number)

order number 124 is delivered second/last in this route - dlvr_seq_num is 2

In this case output format would be as below

order_id	lng	lat	depot_lat	depot_lng	dlvr_seq_num
123	126.59037	43.85294	43.871	126.5661	1
124	126.58835	43.85041	43.871	126.5661	2

For each of the input datasets append a column **dlvr_seq_num (which represents delivery sequence number)** at the end and write to csv

dlvr_seq_num starts from 1 and ends at n in sequence of delivery (if there are n orders to be delivered)

1. part_a_output_dataset_1.csv (all columns of part_a_input_dataset_1 + additional column with name (**dlvr_seq_num**))
2. part_a_output_dataset_2.csv (all columns of part_a_input_dataset_2 + additional column with name (**dlvr_seq_num**))
3. part_a_output_dataset_3.csv (all columns of part_a_input_dataset_3 + additional column with name (**dlvr_seq_num**))
4. part_a_output_dataset_4.csv (all columns of part_a_input_dataset_4 + additional column with name (**dlvr_seq_num**))

5. part_a_output_dataset_5.csv (all columns of part_a_input_dataset_5 + additional column with name (**dlvr_seq_num**))

Write the distance travelled on the best route for each of the dataset to file 6. part_a_best_routes_distance_travelled.csv

- a. Example format of the csv

Dataset	Best Route Distance
part_a_input_dataset_1	16 kms
part_a_input_dataset_2	18 kms
part_a_input_dataset_3	20 kms
part_a_input_dataset_4	22 kms
part_a_input_dataset_5	30 kms

7. part_a_details.txt

1. High level explanation of the algorithm (If it's a standard algorithm/process then you can just write the algorithm name)
2. Algorithm time complexity and space complexity. **If you have used different algorithms for different datasets please write a high level explanation about all of them and their corresponding time/space complexity details**

You have to submit link to the GitHub Repo.

1. Structure of a git repo should look like below **(This structure covers for both part A and part B problems)**
    - a. **walmart-sct-hackathon-round-1** (repo name)
        - i. **code**
            - 1. **part_a**
                - a. <Under part_a directory all code related to part a problem should be present>
            - 2. **part_b**
                - a. <Under part_b directory all code related to part b problem should be present>
        - ii. **input_datasets**
            - 1. **part_a**
                - a. <Under part a there should be 5 input datasets>
            - 2. **part_b**
                - a. <Under part b there should be 2 input datasets>
        - iii. **output_datasets**
            - 1. **part_a**
                - a. <Under part a there should be 5 output datasets + **part_a_best_routes_distance_travelled.csv + part_a_details.txt**> - **Total 7 files**
            - 2. **part_b**
                - a. <Under part b there should be 2 output datasets + **part_b_routes_distance_travelled.csv + part_b_details.txt**> - **Total 4 files**
        - iv. **run_script_part_a.txt** (this file should just contain the command to trigger your code main program/script. This command should work when you are inside the root/repo folder.
        - v. **run_script_part_b.txt** (this file should just contain the command to trigger your code main program/script. This command should work when you are inside the root/repo folder.
            - 1. Example content of run_script_part_a.txt:

```
python code/part_a/run.py  input_datasets/part_a/part_a_input_dataset_1.csv
```

            - 2. Example content of run_script_part_a.txt:

```
python code/part_b/run.py  input_datasets/part_b/part_b_input_dataset_1.csv
```

            - 3. **Based on the choice of your programming language write its corresponding trigger command in run_script_part_a.txt , run_script_part_b.txt**

Evaluation Criteria:

1. For each dataset the following metrics would be considered for evaluation
    - a. Distance travelled by the vehicle
        - i. For part_a_input_dataset_1.csv
            - 1. **Best delivery route/path total distance travelled should be less than 4.5 kms**
        - ii. For part_a_input_dataset_2.csv
            - 1. **Best delivery route/path total distance travelled should be less than 10 kms**
        - iii. For part_a_input_dataset_3.csv
            - 1. **Best delivery route/path total distance travelled should be less than 19 kms**
        - iv. For part_a_input_dataset_4.csv

1. **Best delivery route/path total distance travelled should be less than 22 kms**

v. For part_a_input_dataset_5.csv

1. **Best delivery route/path total distance travelled should be less than 38 kms**

b. Run time complexity, Space complexity, Actual execution run time of the code submitted

2. To minimise the route cost (finding the best delivery route) any external libraries should **NOT** be used. **Route Optimisation algorithm should be implemented on your own. However for data pre-processing / post-processing any external libraries or in built libraries can be used**
3. First Submitted team(s) would be shortlisted if there are multiple teams with same/similar results. (**Keep checking in / committing the code to the git hub repo as soon as you are done with some part of the problem statement**)

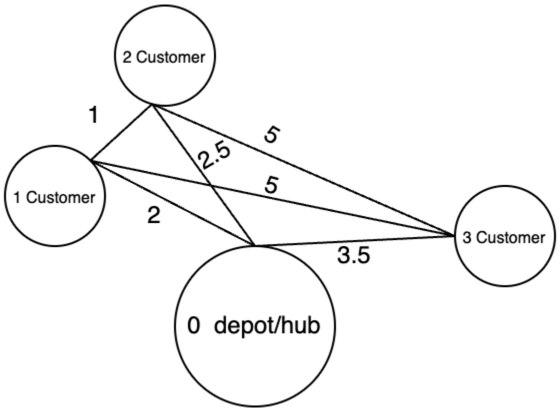
part B: Find the Best Delivery Route using 2 vehicles

Similar to the above problem , lets say we have a limit on number of orders a vehicle can deliver (Vehicle Capacity) . In that case we need to use multiple vehicles to deliver all the orders.

Multiple Vehicles Delivery Process Rules:

1. Each Vehicle will start from depot.
2. Each Vehicle will drive to all the customer locations **on its route** and deliver the package.
3. Each Vehicle will return back to the depot from the last customer delivery location on its route.
4. Each Vehicle will visit every customer location on its route only once.
5. **Each Vehicle can deliver orders only up to its maximum capacity**

Example delivery process/network represented as a Graph with its description



Nodes:

0: depot/hub node

1,2,3: customer nodes

Edges:

Number on each edge represents the distance between the two nodes of the edge in kms.

1. distance from 0 (depot) to 1 (customer) is 2 kms.
2. distance from 0 (depot) to 2 (customer) is 2.5 kms.
3. distance from 0 (depot) to 3 (customer) is 3.5 kms.
4. distance from 1 (customer) to 2 (customer) is 1 kms.
5. distance from 1 (customer) to 3 (customer) is 5 kms.
6. distance from 2 (depot) to 3 (customer) is 5 kms.

Number of Vehicles : 2

Let's say Maximum number of orders each vehicle can deliver : 2 customer orders

Possible delivery routes examples:

Every Vehicle always starts at 0 (depot/hub) and ends at 0 (depot/hub)

1. **Solution 1** :
   a. Vehicle 1 starts at 0 , then visits 2, then 3, and returns back to 0 (Since maximum capacity is 2 , it can only deliver 2 orders) Distance travelled if Vehicle 1 takes this route is 11 kms (2.5 (0- >2) + 5 (2- >3) + 3.5(3->0)))
   b. Vehicle 2 starts at 0, visits 1 and returns back to the 0. Distance travelled if Vehicle 2 takes this route is 4 kms (2 (0- >1) + 2 (1- >0))
   c. Total distance travelled across 2 vehicles to deliver the orders is 15 kms (Vehicle 1 - 11 kms + Vehicle 2 - 4 kms)
2. **Solution 2** :
   a. Vehicle 1 starts at 0 , then visits 1, then 2, and returns back to 0 (Since maximum capacity is 2 , it can only deliver 2 orders) Distance travelled if Vehicle 1 takes this route is 5.5 kms (2 (0- >1) + 1 (1- >2) + 2.5(2->0)))
   b. Vehicle 2 starts at 0, visits 3 and returns back to the 0. Distance travelled if Vehicle 2 takes this route is 7 kms (3.5 (0- >3) + 3.5 (3- >0))
   c. Total distance travelled across 2 vehicles to deliver the orders is 12.5 kms (Vehicle 1 - 5.5 kms + Vehicle 2 - 7 kms)

Solution 2 is better than Solution 1 in this case.

Problem Statement:

Find the best route you think each vehicle should take

- which minimises the total distance travelled across all the vehicles
- delivers all the customer orders following the **Multiple Vehicles Delivery Process Rules** mentioned above
- **If there are multiple solutions possible , output any one of the those solutions**

Input Datasets:

2 datasets

1. part_b_input_dataset_1.csv (30 customer orders + 1 depot/hub + **2 vehicles**)
   a. **Each Vehicle can deliver maximum 20 customer orders**
2. part_b_input_dataset_2.csv (50 customer orders + 1 depot/hub + **2 vehicles**)
   a. **Each Vehicle can deliver maximum 30 customer orders**

Input Dataset metadata would be the same as part A

Output Datasets:

Example Input dataset would look like:

order_id	lng	lat	depot_lat	depot_lng
123	126.59037	43.85294	43.871	126.5661
124	126.58835	43.85041	43.871	126.5661
125	126.56893	43.81414	43.871	126.5661
126	126.56897	43.81418	43.871	126.5661

Lets say your best solution is

Vehicle 1 delivery route : depot - > 123 - > 124 - > depot.

Vehicle 2 delivery route : depot - > 125 - > 126 - > depot.

In this case output format would be as below

vehicle_num - 1,2 (We would be using only two vehicles in this question. This is an identifier for the Vehicle)

dlvr_seq_num starts from 1 and ends at k if there are k orders to be delivered by a vehicle on its route

dlvr_seq_num starts from 1 for every vehicle

order_id	lng	lat	depot_lat	depot_lng	vehicle_num	dlvr_seq_num
123	126.59037	43.85294	43.871	126.5661	1	1
124	126.58835	43.85041	43.871	126.5661	1	2
125	126.56893	43.81414	43.871	126.5661	2	1
126	126.56897	43.81418	43.871	126.5661	2	2

For each input dataset append columns **dlvr_seq_num, vehicle_num** at the end and write to csv

1. part_b_output_dataset_1.csv (all columns of dataset_1 + additional two columns (**dlvr_seq_num, vehicle_num**))
2. part_b_output_dataset_2.csv (all columns of dataset_2 + additional column with name (**dlvr_seq_num, vehicle_num**))

Write delivery route distance travelled across vehicles for each dataset in a different file 3. **part_b_routes_distance_travelled.csv**

a. Example contents of the csv

Dataset	**Vehicle 1 Route Distance**	**Vehicle 2 Route Distance**	**Total distance travelled across vehicle to deliver all orders**
part_b_output_dataset_1	16 kms	12 kms	28 (16 + 12)
part_b_output_dataset_2	13 kms	18 kms	31 (13 + 18)

4. part_b_details.txt

1. High level explanation of the algorithm used , its time complexity and space complexity.
2. If you have used different algorithms for different datasets please write a high level explanation about all of them and complexity details

Evaluation Criteria:

1. For each dataset the following metrics would be considered for evaluation
   a. Total distance travelled across vehicles for each dataset
      i. **For part_b_input_dataset_1.csv**
      ii. **For part_b_input_dataset_2.csv**
   b. Run time complexity, Space complexity, Actual execution run time
2. To minimise the route cost any external libraries should **NOT** be used. **Route Optimisation algorithm should be implemented on your own.**
3. First Submitted team(s) would be shortlisted if there are multiple teams with same/similar results.

Reference Paper : https://arxiv.org/abs/2306.10675

Reference Hugging Face Dataset : https://huggingface.co/datasets/Cainiao-AI/LaDe

Dataset is derived from the above LaDe hugging face dataset

For any queries reach out to kedarnath.reddy@walmart.com , Balivada.Ramkiran@walmart.com