MisMash Hackathon Presentation

MathGuys

Team Members:

1. Harish G
2. Midhilesh E

Problem Statement

Route Optimization and Visualization for Sales Vehicles

What is route optimization?

1. Vehicle Routing Problem (route optimization) is a NP hard problem (i.e the time to compute solution increases exponentially with in size of problem).
2. VRP problem is defined as problem of finding the optimal routes of delivery or collection from one or several depots to a number of cities or customers, while satisfying capacity and time constraints.
3. It is a integer programming problem (i.e the solution space is set of all integers).

Problem formulation

1. Objective function:

   Minimize route distance + Cost of choosing the vehicles + Travel time

2. Constrains:
   a. Travel time for each vehicle < max_travel_time
   b. Distance travelled by each vehicle < max_travel_distance
   c. Capacity at each point < vehicle_capacity
   d. A node cannot be present in two or more routes.
   e. Number of distribution point >=1
   f. A route consist of only one vehicle.

Solution Idea

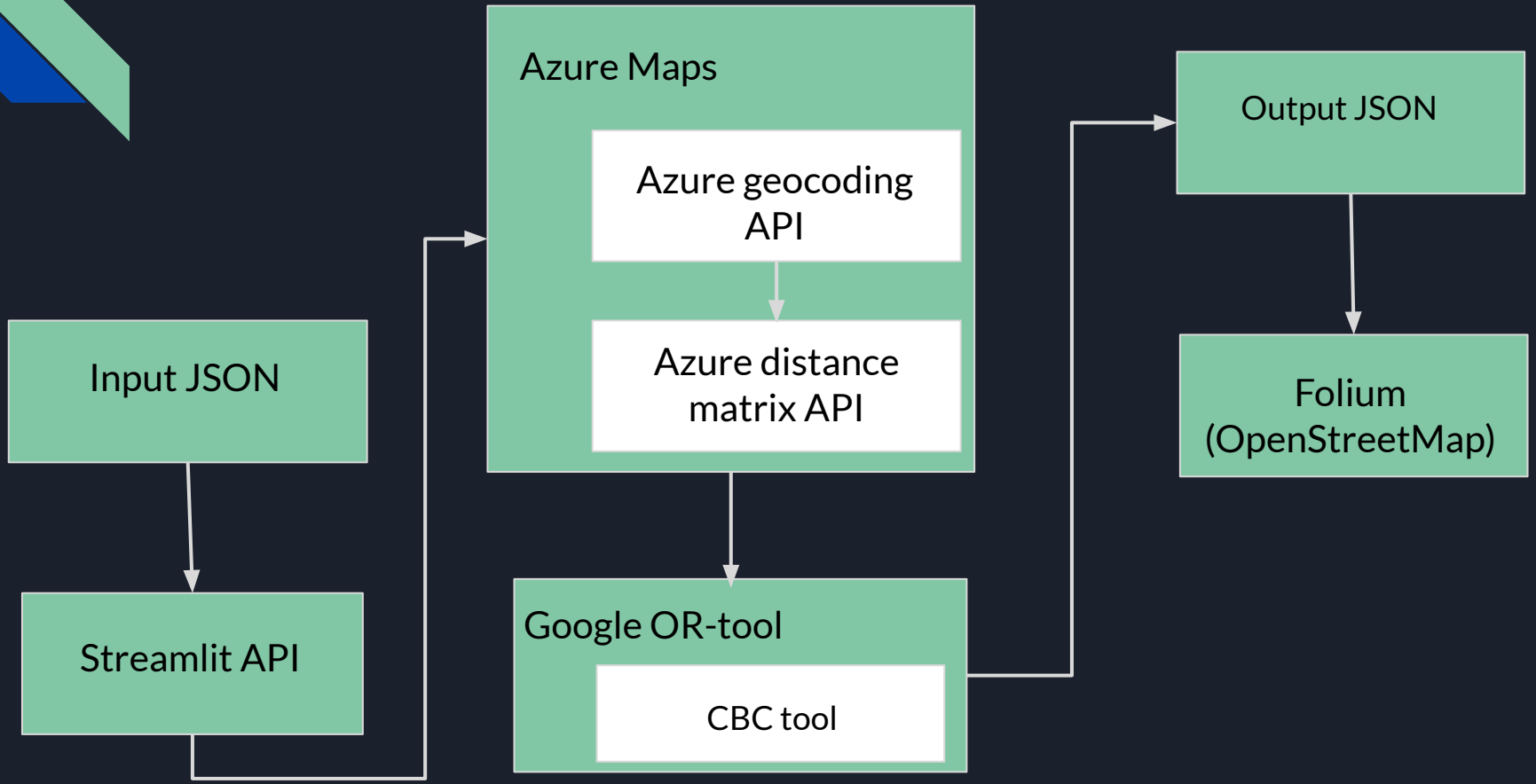
Branch and bound method

1. This is an optimization technique used when both greedy and dynamic programming fails.
2. The general idea behind branch and bound is to split the generate children and choose best children and again split until an optimal solution is reached.

CBC (Coin Branch Cut) tool

1. Google developed OR tool which is wrapper for CBC tool.
2. Using this OR tools the constraints and objective function were formulated.
3. This formulated constrained optimization problem is solved using CBC tool.
4. The tool choses cheapest cost point and branches to find the near optimal solution for the objective function.

Workflow


Azure Maps
Azure geocoding API
Azure distance matrix API
Output JSON
Input JSON
Folium (OpenStreetMap)
Streamlit API
Google OR-tool
CBC tool

Input JSON format

```
{
"distribution_pts":[
   {
   "address": #Location address of distribution point (data type: string),
   "vehicle_capacity": #Contains the capacity of all the vehicles present in the distribution point
   (data type: list),
   "vehicle_costs": #Contains the cost of each vehicle (i.e cost of diesel needed for each vehicle to cover 1Km
   (data type: list),
   "vehicle_speed": #Contains the speed of each vehicles in (Km/hr) (data type:list),
   "max_time": #Contains the maxiumum time a vehicle can travel,(data type:list)
   "max_path_length": #Contains maximum distance a vehicle can travel (data type:list)
   }
   ]
   "delivery_pts":[
   {
   "address": #Location address of distribution point (data type: string),
   "demand": #Demand needed at each location (data type:float or int)
   }
   ]
}
```

Output JSON

```
{'vehicle':
 {'v0':
  {'route_path': #Contains the delivery points and distribution points present in this vehicle route,
  'distance': #Contains the pair wise distance between each point present in this vehicle route,
  'capacity': #Contains the capacity of this vehicle,
  'cost': #Contains the cost of each vehicle (i.e cost of diesel needed for each vehicle to cover 1Km),
  'demand': #Contains demand for each points present in the route,
  'route_load': #Total load carried by this vehicle on this route,
  'route_distance': #Total distance Covered by thix vehicle,
  'travel_time': #Time taken by this vehicle to complete this route (route path)},
  'Total_distance': #Total distance travelled by all the vehicles present in the distribution points,
  'Total_load': #Total load carried by all the vehiles present in the distribution points}
  }
 }
}
```

Workflow explanation

1. Streamlit UI has been developed to receive the input JSON file.
2. With this JSON file the location names were extracted and Azure geocoding API present in Azure maps which return latitude,longitude pair for those location.
3. With the set of [longitude, latitude] values distance matrix was duilt using Azure distance matrix API.
4. Using other information about the vehicles and the distance matrix, the developed branch and bound algorithm using Google OR tools the optimal route is returned as JSON file.
5. The output JSON file was used to plot the route along with other information regarding the route such as vehicle capacity for that route, route distance were **displayed by clicking the nodes and path**

Features added

1. Optimization algorithm runs for 2 minutes with ability to remove nodes from route path such that a feasible solution can be obtained if no solution exist by adding all nodes.
2. Map visualization were fed with all information about the optimal route generated.

   “Click the node and route edge to get more information”

Sample map output