

Delivery Route Optimization

Team - The Quad

Madhumitha Prabakar Dinesh Balasubramaniam Kandan Muthukumar Kavitha Thiagarajan

The Quad - Delivery Route Optimizer

- Route optimization service providing APIs for small retailers and logistics providers to consume
- APIs return optimal routes based on
 - criteria such as distance (or) duration
 - profile such as driving etc.
- Built on <u>OSRM</u> (Open Source Routing Machine) that is backed by OpenStreetMap data

Accessing the APIs

The Delivery Route Optimizer service is deployed in GCP (Google Cloud Platform).

APIs can be accessed <u>here</u>.

User personas

User Persona 1

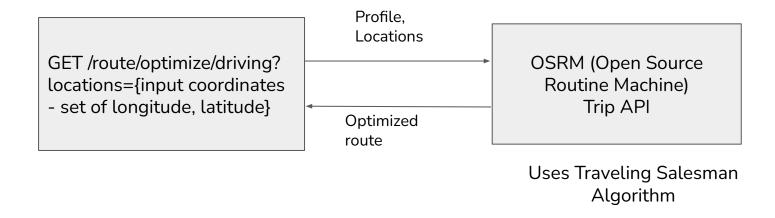
Gupta Kirana Store, is a small retailer, who is **not connected** to any **logistic provider**.

The store delivers products to the customers on their own i.e. they have their own delivery person employed.

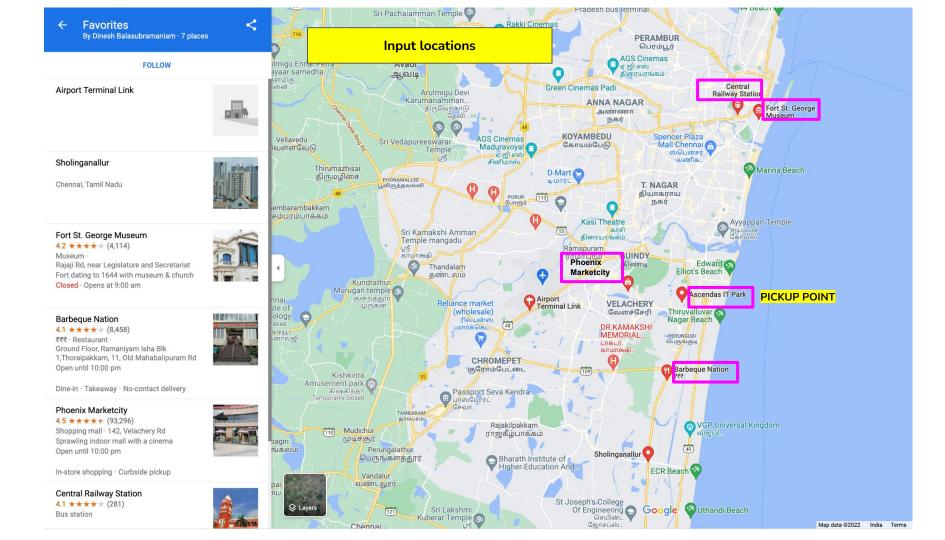
On any day, the store receives a set of orders which are handed over to the employed delivery person who would want to find out an optimal route between the set of delivery locations so that the cost or time is subsidized.

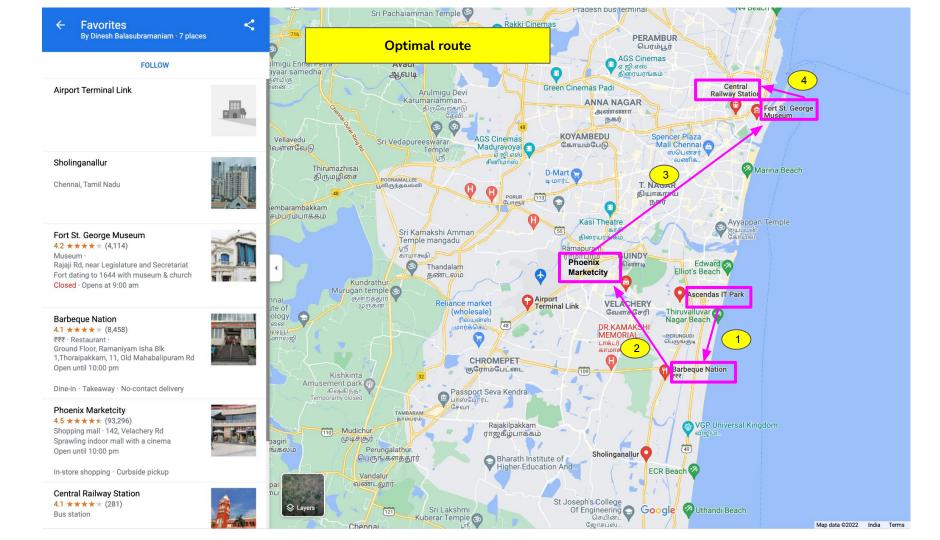
This use case is - single delivery agent, single pickup point, multiple delivery points

Basic Route Optimization API



Demo of Basic Route Optimization API





User Persona 2

Bharat Stores, is a retailer, who is predominantly online based.

The store receives online orders and runs its **own logistics service** to meet the high delivery needs.

The logistics service associated with Bharat Stores would like to group the orders optimally and assign them to different delivery agents such that each delivery agent picks up multiple orders which have close delivery points.

This use case is - multiple delivery agents, single pickup point, multiple delivery points

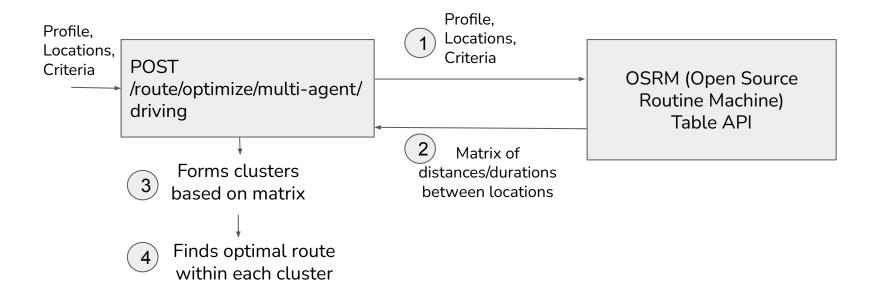
Example of User Persona 2

Delivery Points **A**, **B** and **C** are nearer to each other. Delivery Points **D** and **E** are closer to each other but are farther from A, B and C.

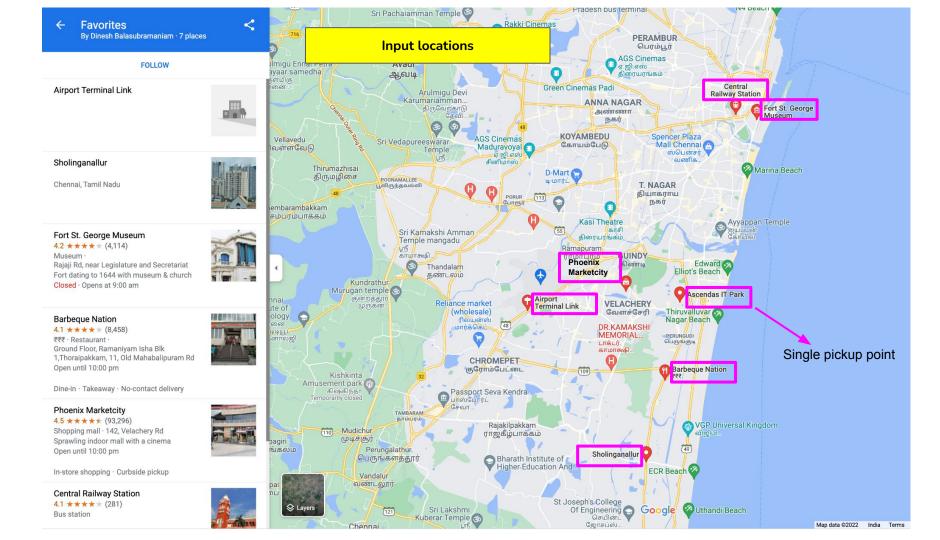
In this case, the logistics service would want to group A, B and C orders and assign them to Delivery Person **P1**. Similarly, they would want to group D and E and assign them to Delivery Person **P2**.

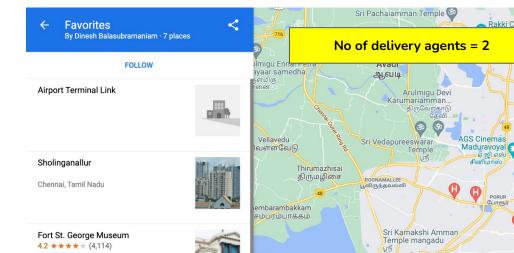
Also, between A, B and C, the delivery person P1 would want to find an **optimal route** to travel and deliver the orders.

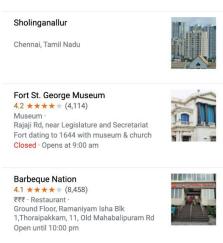
Multi-Agent Route Optimization API

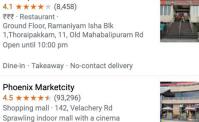


Demo of Multi-Agent Route Optimization API

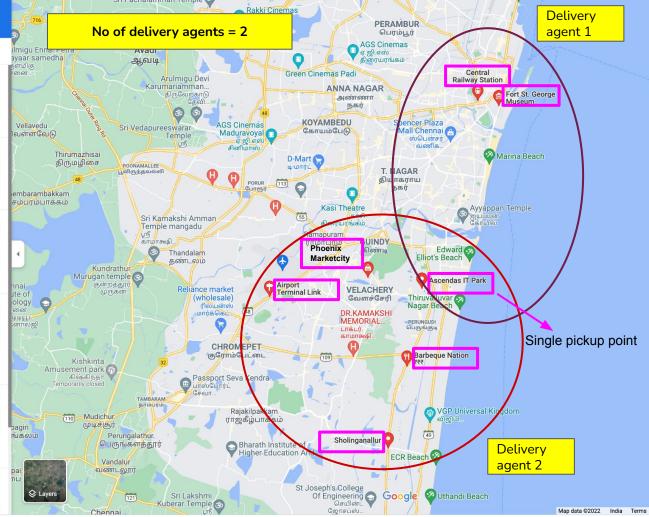












Pradesh bus terminal



FOLLOW

Airport Terminal Link



Sholinganallur

Chennai, Tamil Nadu



Fort St. George Museum

4.2 **** (4,114)

Museum ·

Rajaji Rd, near Legislature and Secretariat Fort dating to 1644 with museum & church Closed · Opens at 9:00 am



Barbeque Nation

4.1 **** (8.458)

₹₹₹ · Restaurant ·

Ground Floor, Ramaniyam Isha Blk 1,Thoraipakkam, 11, Old Mahabalipuram Rd Open until 10:00 pm



Dine-in · Takeaway · No-contact delivery

Phoenix Marketcity

4.5 **** (93,296)

Shopping mall · 142, Velachery Rd Sprawling indoor mall with a cinema Open until 10:00 pm



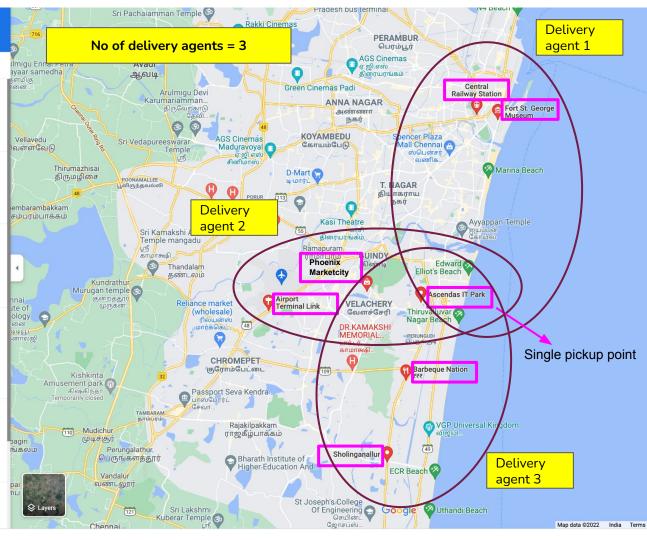
In-store shopping · Curbside pickup

Central Railway Station

4.1 **** (281)

Bus station





User Persona 3

I-Delivery, is a **logistics provider**, who picks up orders from various retailers and delivers them doorstep to their respective customers.

Any delivery person working in I-Delivery would want to find an optimal route to pick up orders from various shops and deliver them to their customers.

This use case is - single delivery agent, multiple pickup points, multiple delivery points

Example of User Persona 3

Given (pickup_1, drop_1), (pickup_2, drop_2), (pickup_3, drop_3),

An optimal route could be - a delivery person picking up an order from pickup_1 and then proceeding to, say, pickup_2 which is nearer to pickup_1.

Then the person finds drop_2 to be closer and hence delivers at drop_2.

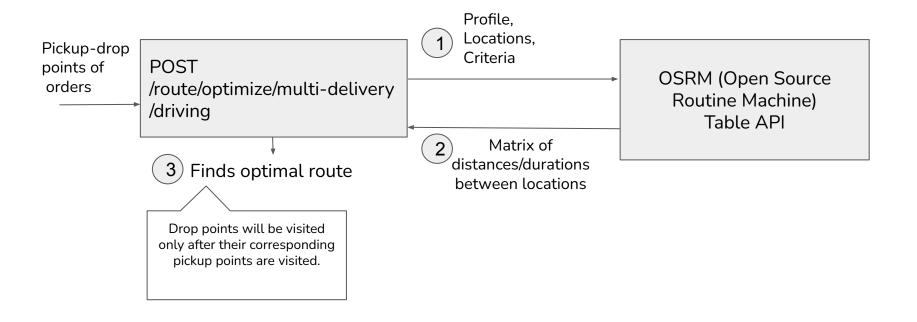
The next closest stop is pickup_3 and hence the person picks up order and finds out that drop_1 is nearer to pickup_3.

Example of User Persona 3 Contd.

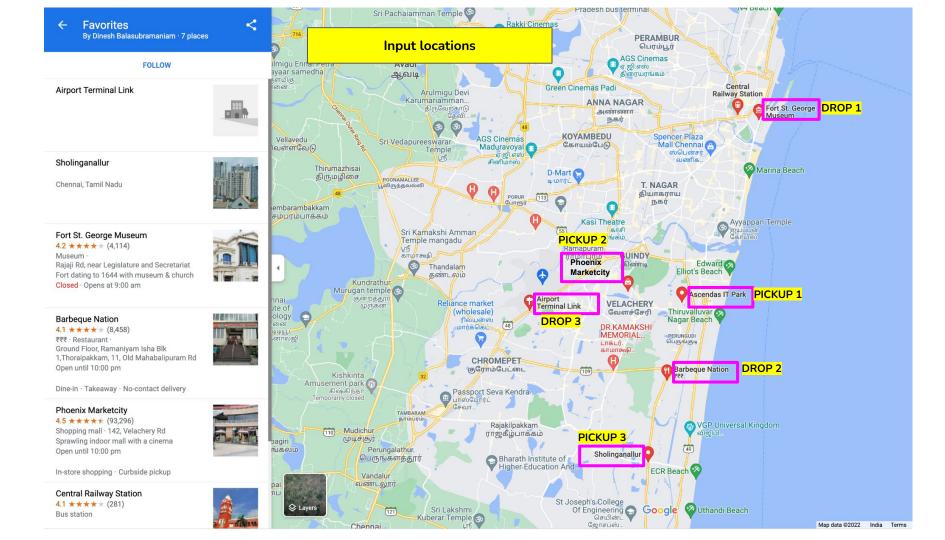
Now after delivering to drop_1, the person finally arrives at drop_3 to deliver.

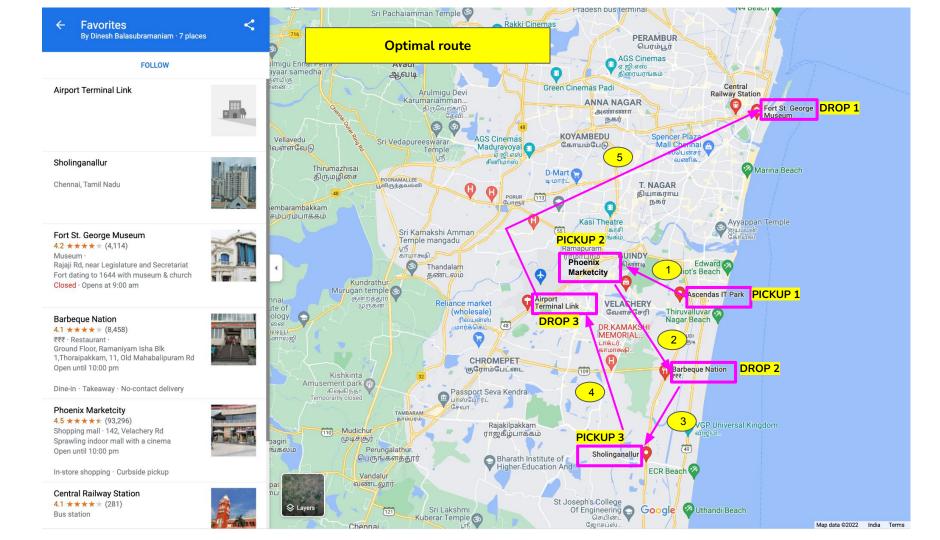
pickup_1 -> pickup_2 -> drop_2 -> pickup_3 -> drop_1 -> drop_3.

Multi-Point Delivery Route Optimization API



Demo of Multi-Point Delivery Route Optimization API



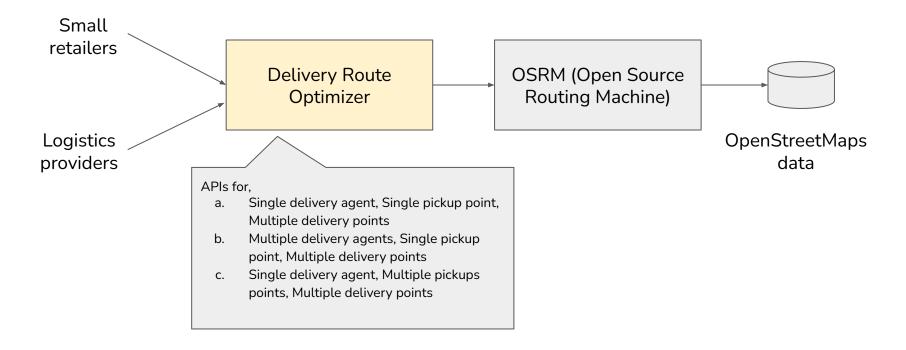


Algorithms Used

- Traveling salesman problem
- K Spanning tree
- Multi-Level Dijkstra (internally by OSRM)

Language used: Java

Architecture Diagram



Future Enhancements

- To support multiple delivery agents picking up from multiple pickup points and delivering to their respective drop points.
- To extract the route optimization algorithms as a Java library that is pluggable by other applications
- Mobile app SDK for small-retailers' and logistics providers' applications, if any, to communicate with our APIs.
- To display the optimized route with driving directions/steps.
- To consider factors such as road closures, water logging etc. while finding optimal routes.

Thank you