Xpressathon Challenge
Geo Intelligence

Team Disruptors
Chirag Hegde, Ayush Yadav, Sushil Adwe

January 27, 2022

1 Introduction

Last mile delivery refers to the last leg of supply chain operations. A product's journey from a warehouse to the doorstep of the end-customer. This last step of the delivery process is most critical and should be optimized for a better user experience and dramatic reduction in operating costs.
We are proposing a Mixed Integer Programming (MIP) model for allocating shipments to the nearest service centers while keeping in mind the capacity constraints.
Further we formulate a Greedy heuristic and compare the performance of both the solutions.

2 Solution Formulation

2.1 Mixed Integer Programming Optimization

We propose a MIP model for the optimal **allocation** of the shipments to the service centers to minimize the cost.
Let the number of warehouses be n and the number of shipments be m. Each warehouse is denoted by i = 1, 2, ... , n and each shipment by j = 1, 2, ... , m.
Let c_{ij} be the cost of delivering a shipment from the i^{th} Warehouse to the j^{th} Shipment location. For simplicity, we take the cost to be the distance between the two locations.

Let us now introduce the **Decision Variable** x_{ij}.
x_{ij} takes the value 1 if the i^{th} warehouse delivers the j^{th} shipment and 0 if not.
Since each warehouse has a capacity on the number of deliveries it can perform in a day, we want $\sum_{j=1}^{m} x_{ij} \leq cap_i$ where cap_i is the capacity of the i^{th} warehouse.
Now, each shipment can have two outcomes, it being assigned to a depot or it being shifted to the next day for delivery. Hence we get another constraint $\sum_{i=1}^{n} x_{ij} \leq 1$.
Of course if we allow a shipment to be un-allocated for no penalty there will be no allocations! Hence for every shipment that remains un-allocated we penalise the optimizer to ensure that minimal shipments are left un-allocated.

The cost function for our model is

$$\boxed{\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}.c_{ij} + \lambda.(1 - \sum_{i=1}^{n} x_{ij})}$$

Thus the final model looks like:

$$min_x \left(\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}.c_{ij} + \lambda.(1 - \sum_{i=1}^{n} x_{ij}) \right)$$

Subject To:

$$\lambda \in penalty$$

$$\sum_{j=1}^{m} x_{ij} \leq cap_i \qquad \forall i$$

$$\sum_{i=1}^{n} x_{ij} \leq 1 \qquad \forall j$$

$$x_{ij} \in \{0, 1\}$$

This was solved using Google's OR-Tools with the CBC Solver backend (both Open Source).

2.2 Performance Evaluation

We ran our model on some test data and the results were visualized.

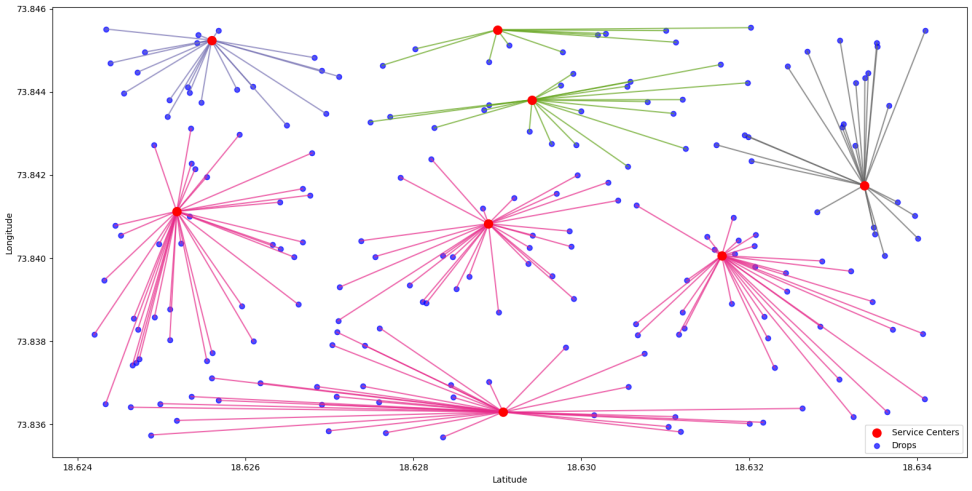

Figure 1: Low density of Shipments

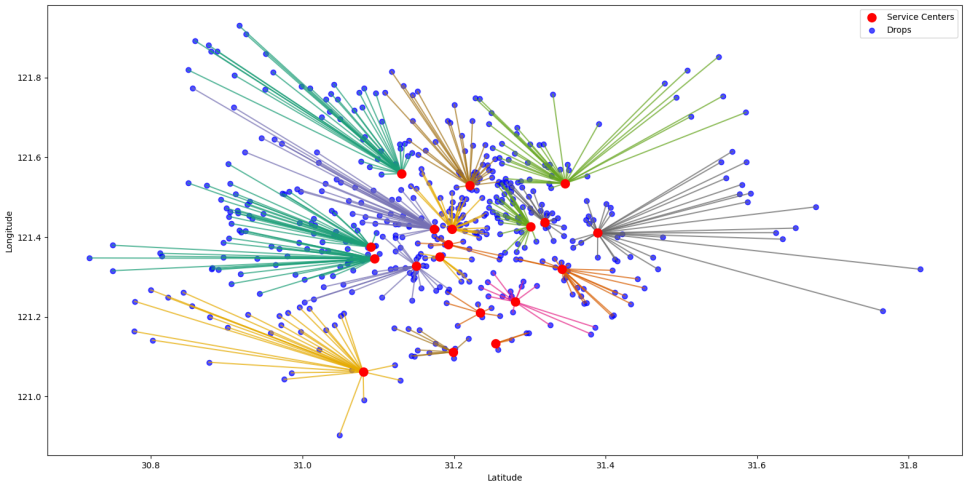

Figure 2: High density of Shipments

2.3 Optimal Route Formulation

Once each shipment is allocated to a warehouse, we need to optimize our delivery process with the following factors in mind:

- Minimize the total distance travelled
- Ensure that packages do not exceed vehicle capacity
- Minimize delivery time

We tackled this problem by modelling each warehouse's allocations as a Capacitated Vehicle Routing Problem (CVRP).

To deivise the optimal routes, the following information is necessary for each warehouse:

- The number of delivery vehicles
- The maximum capacity of each vehicle
- The weight of each shipment

In our performance evaluations, we made the following assumptions for simplicity but these can be easily extended to match the real life scenarios:

- The number of vehicles at each warehouse is $\left(\frac{Numberofshipments}{30}\right)$
- The capacity of each vehicle is 25 units
- The weight of each shipment is 1 unit

2.4 Visualizing the final output

The results our program returned on sample test data:

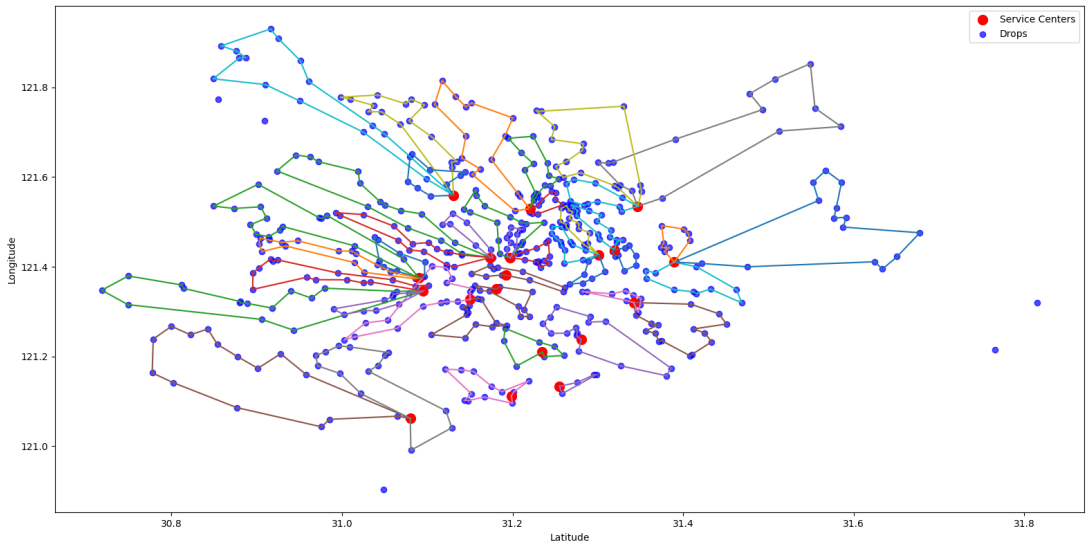

Figure 3: Final Routes returned by the algorithm

3 Execution Workflow

Let us assume a district with 1000 shipments scheduled for tomorrow and a central distribution facility which holds all the shipments before distributing them to the service centers.

The execution can be split into two phases:

3.1 Allocation of the shipments to the service centres

The central facility will run the allocation algorithm which will optimally allocate each shipment to a service center to minimise the **cost** and **time latency**. Once the shipments are allocated, they are transported to the service centers for further deliveries.

3.2 Route planning of these shipments from each service centre

Each service center now has the knowledge about the location, quantity and weight of each shipment it is supposed to deliver.
They will now run the Route Planning algorithm to optimally form routes for the deliveries keeping in mind the capacity constraints of each available delivery vehicle.

4 Heuristics

4.1 Greedy Algorithm

A straightforward greedy algorithm would be to sort all the shipment - service center location and go on fulfilling them according to the capacities. But this solution fails for many edge cases (Figure 4).
We formulate another greedy algorithm which even though does not perform better than the MIP model, performs much better than the conventional greedy approach.
We describe the algorithm below:

- Initially assign each shipment to its nearest service center irrespective of the capacity of the service center.
- Repeat the following steps until the number of shipments allocated to each service center is less than it's capacity (It can be proved that the number of iterations are bound by the number of service centers):
  - Traverse over each service center and for each service center that has been overloaded:
    * Find the number of overloaded shipments = n.
    * For each shipment assigned to this service center, we find the least possible additional cost that will be incurred if this shipment is allocated to another service center.
    * We transfer n shipments with the least transfer costs to their next nearest neighbour service center.

This modified greedy model gives a decent solution even on edge cases like below where conventional greedy algorithms might fail.

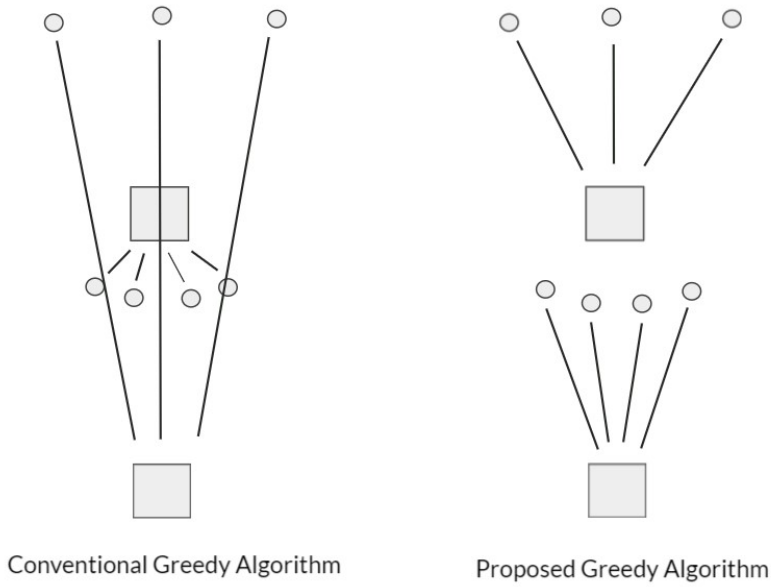

Figure 4: Edge case where conventional greedy solution fails

4.2 Performance Comparison

The following figures compare how our greedy model compares with the MIP solution on a test Data set of 18 service centers and 500 shipments spread across the city.

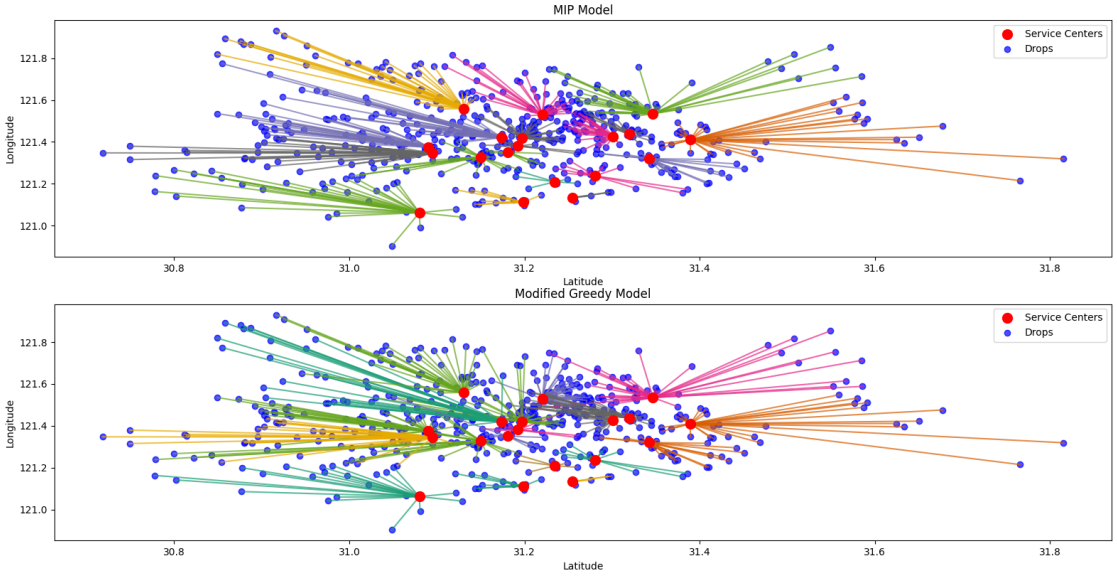

Figure 5: Cost Comparison
MIP: 5662.81 km — Greedy: 5921.69 km