

## The simulation process is as follows:

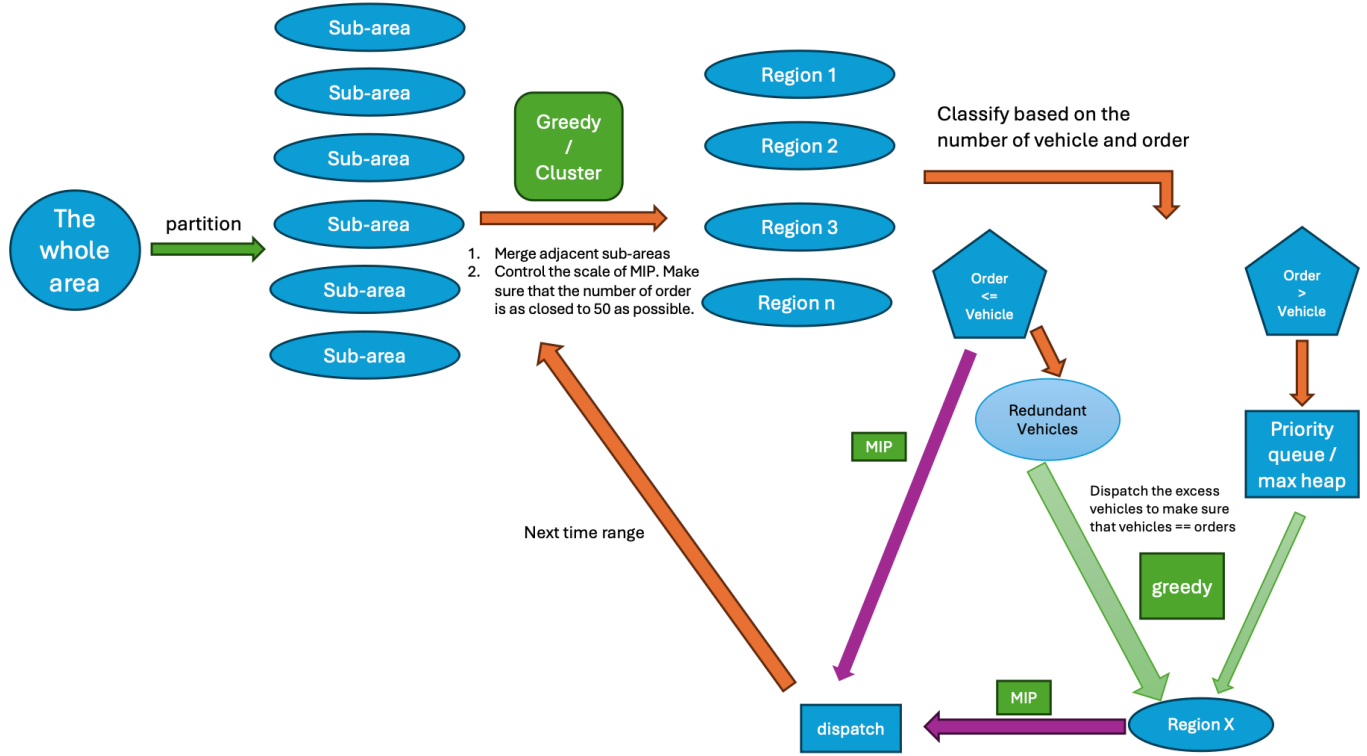
1. initialize all of vehicles in the city, generate the user requests based on the record from 滴滴 / NYC taxi, and init the map of Chengdu/ New York City.
2. in every time range (like 2 seconds), update vehicle's request and current situation (like, current position, if accept request, if accept car sharing, etc.).
3. Calculate the waiting time to pickup for every user request and the money they spend on the trip.

## The rule for vehicle-order:

1. One order is only for one user.
2. One car can accept up to two orders.
3. Three behaviors:
  1. Two people simultaneously hail a ride within a certain time range and decide to share the ride. The car will pick up Person A first, then Person B, and drop off Person A first, followed by Person B.
  2. A person willing to share a ride, Person A, is already in the car and the car has started the trip. Suddenly, a new carpool request from Person B is received. The car will then go from its current location to pick up Person B, drop off Person A, and finally drop off Person B.
  3. A person takes a ride from start to finish without any carpooling.

## The workflow:

1. Divide the whole city map into multi sub-areas.
2. In every time range:
  1. Based on the current relationship between requests and vehicles, utilize the greedy algorithm or cluster algorithm, and merge the adjacent sub-areas into one region until the number of order is as closed to 50 as possible (Control the size of MIP)
  2. Classify these regions into two types: 1. Number of order is greater than number of vehicles 2. Number of order isn't greater than number of vehicles.
    1. for type 2: MIP
    2. for type 1: dispatch the excess vehicles into busy area until the number of vehicle equals to the number of order, by using **greedy algorithm/LLM**. (LLM is not suitable for this part). Then MIP.
  3. Solve every MIPs with solver/LLM
  4. After determining the decision for every vehicles, running the simulation process.
3. Using the cost for every user and the waiting time as the metrics. And compare the solving time of MIP Solver and LLM.



## Mathematical Formulation for Dispatch

Since the cost of calculating the distance using dijkstra algorithm is very expensive and unrealistic, we only calculate the manhattan distance when we build the MIP model. In the simulation process, using dijkstra algorithm to simulate car move.

### Objective Function

The objective is to minimize the total Manhattan distance between vehicles and users, considering all segments of the vehicle paths:

$$\min \sum_{i=1}^{n_1} \sum_{j=1}^m x_{ij} \cdot d_{ij} + \sum_{i=1}^{n_1} \sum_{j=1}^m \sum_{k=1, k \neq j}^m y_{ijk} \cdot (d_{ij} + d'_{jk}) + \sum_{i=1}^{n_2} \sum_{j=1}^m z_{ij} \cdot d''_{ij} \quad (1)$$

where:

- $x_{ij}$  is a decision variable that indicates whether empty car  $i$  is assigned to user  $j$  (1 if yes, 0 if no).
- $y_{ijk}$  is a decision variable that indicates whether empty car  $i$  is assigned to pick up user  $j$  and then user  $k$  (1 if yes, 0 if no).
- $z_{ij}$  is a decision variable that indicates whether car  $i$  with one passenger willing to share is assigned to user  $j$  (1 if yes, 0 if no).

- $d_{ij}$  is the Manhattan distance between vehicle  $i$  and user  $j$ .
- $d'_{jk}$  is the Manhattan distance between user  $j$  and user  $k$ .
- $d''_{ij}$  is the Manhattan distance between vehicle  $i$  which has one order and user  $j$ .

## Constraints

the number of empty vehicles \* 2 + the number of vehicles with one sharing order <= the number of coming order

1. Every user is assigned one vehicle

$$\sum_i x_{ij} + \sum_i \sum_{k, j \neq k} (y_{ijk} + y_{ikj}) + \sum_{i'} z_{i',j} = 1, \forall j \quad (2)$$

2. Every empty vehicle pickup at most two people

$$\sum_j x_{ij} + \sum_j \sum_{k \neq j} y_{ijk} \leq 1, \forall i \quad (3)$$

3. Every vehicle which has one passenger pickup at most one people

$$\sum_j z_{ij} \leq 1, \forall i \quad (4)$$

4. Binary variable

$$z_{i',j} = 0, 1, \forall i', j \quad (5)$$

$$x_{ij} = 0, 1, \forall i, j \quad (6)$$

$$y_{ijk} = 0, 1, \forall i, j, k \quad (7)$$