

Discussion 5

Routing Optimization System

Problem

Consider that we have an environment having a set of Riders $R = \{R_1, R_2, R_3, \dots, R_N\}$ and some Restaurants $P = \{P_1, P_2, P_3, \dots, P_M\}$, also known as Pickup locations in our context, and some Orders $O = \{O_1, O_2, O_3, \dots, O_K\}$, which are being received continuously, one after the other, at the hub. The hub is the central point where all orders are received and the hub has to decide which rider should be assigned what order, such that the overall time to complete all the jobs is minimized. Each order has an impact of t_l minutes on the overall time of a particular rider R_n , and an impact of t_g minutes on the overall completion time of all the jobs.

When an Order O_n arrives at the hub for a pickup location P_m , the hub filters out the riders that are near to the pickup location P_m and for each rider r_i , a fitness value f and the total time rt_i that the rider will take to complete its job after the assignment of the current order, is calculated to find the best possible rider. On the basis of the fitness value of all the riders, the hub will decide which rider is to be assigned what order.

Discussed Solution

- When an order arrives at the hub, it is organized according to the pickup location or the restaurant an order has been received for. Each restaurant has its own order queue and when an order is received, it is added to its respective restaurant's queue.
- The process after organizing the orders is based on the order queue and therefore is the same for every order queue or restaurant. The process after this will occur for every restaurant that has an order.
- For every restaurant, two processes will be carried out periodically at regular intervals, which are described below:
 - **Prediction:** A prediction will be made to guess the expected number of orders that the restaurant might receive in the next hour. This prediction will be made on an hourly basis. By doing so, we can bring more riders near to a restaurant which is going to receive more orders in the coming hours by assigning the current orders to more far away riders.
 - **Filtering:** A filtering process will be carried out every minute to update the Riders subset for that particular restaurant. This Riders subset would contain the riders that could possibly be the optimal one for a particular order. This process will keep updating the Riders subset so that we never run out of riders as we would be dispatching the riders along with the orders and some riders would also have

finished their job. In this way we will also involve them when planning deliveries for the new orders.

- When doing the filtering process, a rider can fall in more than one subsets. If a rider in one subset appears to be the optimal one, and the same one is also the optimal for some other subset, then the rider will be compared for all the orders and whichever is minimizing the rider's job completion time, will be assigned to that rider.
 - After filtering the riders for a pickup location, for every rider, its fitness value with respect to the order, will be calculated. The fitness function would depend on the following features:
 - **Rider:**
 1. Current number of orders assigned.
 2. Total job completion time of the rider after the assignment of the current order (est. time for previous orders + est. time for the current order).
 - **Order:**
 1. Order received time.
 - After calculating the fitness values of all the riders, it would check which rider is getting the job done in less time. The rider with the least fitness value will be chosen as the optimal rider for the job and the order will be assigned to that particular rider.
 - At last, the rider's order count will be updated and again checked. If the order count of that particular rider has reached the maximum limit, the rider will be removed from the restaurant's filtered rider's list.
-

An order arrives at the hub

The order is organized into restaurants

- queues for each restaurants
- order is added to the restaurant's order queue

Foreach restaurant queue

- predict orders in the next hour
- Filter the riders for the restaurant
- - input the predicted order count
- Foreach Rider
 - Foreach order in the queue
 - Calculate riders fitness value w.r.t the order
- Foreach order in the queue
 - check which rider is getting the job done in less time
 - - also check that no order is delivered in more than 45 mins
 - assign the order to the rider with the minimum job completion time
 - check if the riders max order limit is reached
 - - if yes, remove the rider from the list