Discussion 3

Routing Optimization System

Problem

Consider that we have an environment having a set of Riders $\boldsymbol{R} = \{\boldsymbol{R_1}, \boldsymbol{R_2}, \boldsymbol{R_3}, \ldots, \boldsymbol{R_N}\}$ and some Restaurants $\boldsymbol{P} = \{\boldsymbol{P_1}, \boldsymbol{P_2}, \boldsymbol{P_3}, \ldots, \boldsymbol{P_M}\}$, also known as Pickup locations in our context, and some Orders $\boldsymbol{O} = \{\boldsymbol{O_1}, \boldsymbol{O_2}, \boldsymbol{O_3}, \ldots, \boldsymbol{O_K}\}$, 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 $\boldsymbol{t_l}$ minutes on the overall time of a particular rider $\boldsymbol{R_n}$, and an impact of $\boldsymbol{t_g}$ minutes on the overall completion time of all the jobs.

When an Order $\boldsymbol{O_n}$ arrives at the hub for a pickup location $\boldsymbol{P_m}$, the hub filters out the riders that are near to the pickup location $\boldsymbol{P_m}$ and for each rider $\boldsymbol{r_i}$, a fitness value $\boldsymbol{f}$ and the total time $\boldsymbol{rt_i}$, that the rider will take to complete its job after the assignment of the current order, is calculated to find the best possible rider. This process is dynamic i.e. if another order $\boldsymbol{O_{n+x}}$ arrives for the same pickup location, while the earlier one(s) were being processed, the fitness value and the total time to complete the all the jobs, with respect to the new order will also be calculated for all the filtered riders. On the basis of the fitness value of all the riders, the hub will decide which rider is to be assigned what order.

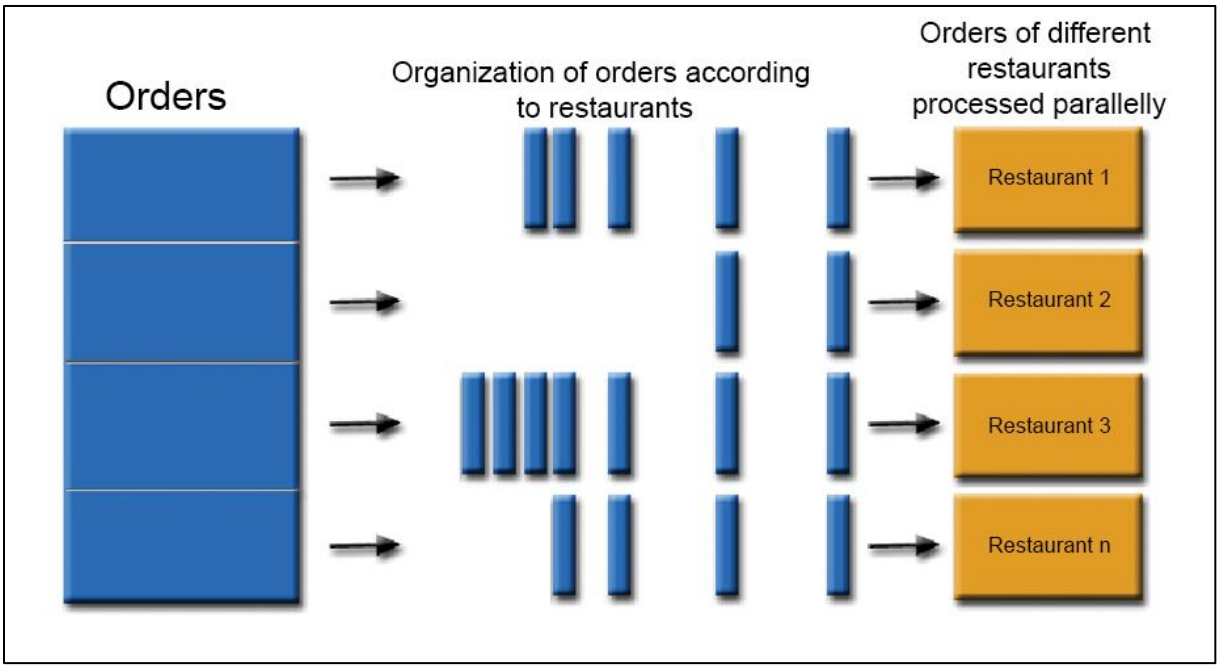
Assumptions

Following are some assumptions for the project.

- The total number of riders is less than the number of orders.
- A rider cannot have more than 5 orders at a time. As for some real businesses, a rider is sent out for deliveries with a maximum of five orders.
- An order can take maximum 45 minutes to be delivered. As in real world situations most restaurants would ask for 45 minutes for the food to be delivered.

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.
- After organizing the orders, we can use parallel computing to process the orders for all the restaurants at the same time.



- The process from here, will occur for every restaurant that has an order.
- The newly received order will be added to the orders queue of its respective restaurant.
- 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 order that the restaurant might receive in the next hour. This prediction will be made on 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 and for the overall job as well. 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 checked with be for all those orders

and whichever is minimizing the overall job completion time(global time), will be assigned to that rider.

- After filtering the riders for a pickup location, for every rider, we will calculate the fitness value with respect to the order. The fitness function would depend on the following features:
    - **Rider:**
        1. Current position of the rider.
        2. Current number of orders assigned.
        3. 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.
        2. Order ready time.
- At this point we would have the fitness values for all the riders and we will be in a position to assign the order to the most optimal rider, but, we will first check if we have any new order in the queue. If we have any new order, we would also check fitness values with the new orders, and then check whether assigning the order to the most optimal rider would get the overall job done in less time or we can assign the orders in such a way that the order is not assigned to the most optimal rider, but still we can get the overall job done in much less time.
- Another problem arises here, i.e. if we are checking the new incoming orders, and the process of receiving orders in dynamic, then there is a possibility that we may never reach to a point of assigning the orders to the riders. To overcome this, we will limit this checking process to 1 minute i.e. the orders received in the next minute will not be checked until all the orders received in the previous minute are assigned to their respective riders.
- After this step, any rider whose maximum limit of orders has reached, will be removed from the Riders subset.
