### **Last Mile Delivery Batching**

### **Problem Statement:**

Description: It is crucial in today’s last-mile delivery ecosystem to optimize for speed,

and cost efficiencies. Smarter algorithms play a crucial role in the ecommerce marketplace

deliveries We need to group/batch the delivery of multiple items to the same rider without

losing time. Here are several scenarios where we need smart operational research algorithms.

Rule # 1:

• Two orders - From the same kitchen.

• To the same customer.

• Ready at the same time (10 mins apart).

• Assign the pick-up to the same rider.

Rule # 2:

• Two orders.

• From two different kitchens (1 km apart).

• To the same customer.

• Ready at the same time (10 mins apart).

• Assign the pick-up to the same rider.

Rule # 3:

• Two orders.

• From the same kitchen.

• To two different customers (1 km apart).

• Ready at the same time (10 mins apart).

• Assign the pick-up to the same rider.

Rule # 4:

• Two orders.

• From two different kitchens (1 km apart).

• To the same customer.

• Ready at the same time (10 mins apart).

• Assign the pick-up to the same rider.

Rule # 5:

• Two orders.

• From two different kitchens (1 km apart).

• To the same customer.

• Ready at the same time (10 mins apart).

• Assign the pick-up to the same rider.

Rule # 6:

• Two orders.

• To the same customer.

• 2nd kitchens pick up on the way to the customer.

• Ready at the time the rider reaches the second kitchen (10 mins apart).

• Assign the pick-up to the same rider.

Rule # 7:

• Two orders.

• 2nd customers drop on the way to the 1st customer (Vice Versa).

• 2nd kitchens pick up on the way to the customer.

• Ready at the same time (10 mins apart or by the time rider reaches the kitchen).

• Assign the pick-up to the same rider.

Rule # 8:

• Two orders.

• From the same kitchen.

• 2nd customers drop on the way to the customer 1st (Vice Versa).

• Ready at the same time (10 mins apart).

• Assign the pick-up to the same rider.

### **Approach:**

* Order Management
* Grouping Orders
* Route Optimization
* Assigning Riders

### **Implementation Details:**

* Order Management:
  + Maintain the orders with attributes such as pickup location, drop-off location, ready time, and duration.
  + Orders are accepted during a specified duration and stored in a queue.
* Grouping Orders:
  + Iterate through the queue and apply rules to group orders that meet specific conditions:
    - Group orders based on rules such as same kitchen, same customer, proximity, and ready time.
* Route Optimization:
  + For each group of orders, calculate the optimal delivery route:
    - Utilize OR-tools for solving the Vehicle Routing Problem(VRP) .
    - Consider factors such as distance and order ready time.
    - Optimize for the shortest route and minimize rider wait times.
* Assigning Riders:
  + Assign grouped orders to available riders:
    - Ensure riders can handle the combined orders within their capacity and time constraints.
    - Riders are implemented as threads.
    - Prioritize assignments based on rider proximity to pickup locations and order urgency.

### **Challenges Faced:**

* + Learning and integrating with Google OR-tools for route optimization.
  + Integrating the given constraints to a predefined algorithm ie. the or-tools.
  + Implementation of threads in python to mimic real time order placement and response.

### **Future Improvements:**

* Machine Learning Integration:
  + Use historical data to predict order volumes and optimize route planning.
  + Implement reinforcement learning for dynamic route adjustments based on real-time conditions.
* Real-Time Tracking and Communication:
  + Enable real-time tracking of orders and riders for better coordination.
  + Implement communication channels between riders, customers, and support staff for seamless operations.
* Scalability and Performance:
  + Optimize algorithms and data structures for scalability to handle large order volumes.