When Routing Meets Recommendation: Solving Dynamic Order Recommendations Problem in Peer-to-Peer Logistics Platforms

Anonymized

No Institute Given

1 DORP Example

Figure 1 provides an simple illustration of the proposed problem. In this example, we assume that the planning horizon are discretized into 10-minute time periods. At 8 : 00, there are a total of 4 unassigned orders in the order list and 2 active drivers. The order list include newly-arrived orders and leftover unassigned orders from the previous time period. Given the two lists, the platform generates a order recommendation list for each driver. Drivers then can decide to select at least one order or reject all orders. Note that *Order 3* is recommended to both drivers. Depending on the assignment rule being used (fastest-finger-first rule is usually adopted in real-world operation), the drivers will be allocated the jobs and their existing delivery schedules will be updated.

2 ARH Algorithm

Algorithm 1 shows the how the ARH generates the recommendation lists in a given decision epoch.

3 DQN Hyper-parameters

The hyper-parameters we used can be seen Table 1.

4 Feature Engineering

The features used for training the xDeepRM model are presented in Table 2.

5 Data

Figure 2 shows the distribution of the location of all orders in one instance i.e one-day scenario. The dataset contains two major components; one data pertain to the order data i.e. the pickup-delivery task with its corresponding features. While the other pertains to the driver i.e the driver's features and previous completed jobs. The exact details of the features used are provided above.

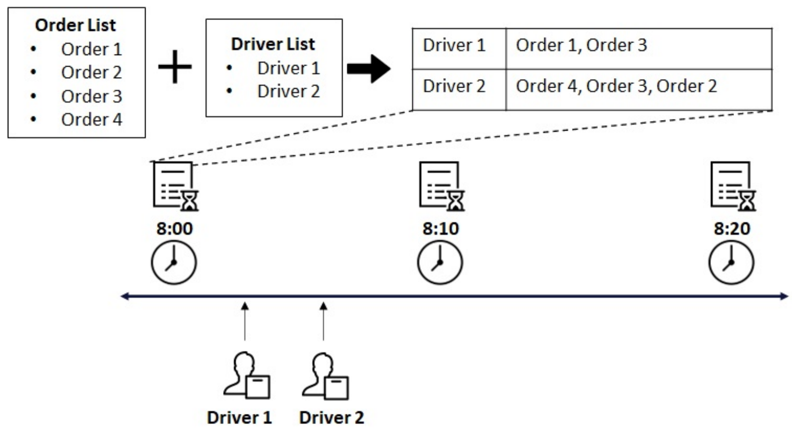

Fig. 1. An illustration of a DORP.

Algorithm 1 Generate recommendation list by ARH with a trained DQN

INPUT: *unallocated_orders*; *drivers*; *current_time*; *max_list*
OUTPUT: *recommendation_list*

initial *recommendation_list* for all *drivers*
$\alpha \leftarrow$ ChooseActionByDQN ($unallocated\_orders$, $drivers$, $current\_time$)
for each $driver_i$ in *drivers* **do**
    $temp\_list = [\,]$
    for each $order_j$ in *unallocated_orders* **do**
        $H_1 \leftarrow$ CalculateRoutingHeuristic ($driver_i$, $order_j$)
        $H_2 \leftarrow$ CalculateXDFM ($driver_i$, $order_j$)
        $Score_{ij} = \alpha H_1 + (1 - \alpha) H_2$
        append $Score_{ij}$ to *temp_list*
    end for
    sort *temp_list*
    $temp\_list = temp\_list[: max\_list]$
    add *temp_list* to *recommendation_list*
end for

To train the xDeepFM model, we split the dataset into two subsets with the proportion of 80% and 20% for training and testing respectively. Meanwhile, to train the DQN in our ARH, we select a slice of the historical data with stable distribution. From Figure 3, we can see the number of orders is not stable in the beginning as the platform just started operating at that period. Thus, we select a slice from the stable part of the dataset to create two subsets of data for training and testing. The training data set includes two-month worth data while the size of the testing data is set at two weeks.

Hyper-parameter	*Setting*
Hidden layer size	$50, 50$
Memory size	2000
Mini-batch size	32
Target network update period	300
Learning rate	0.01
Discounter factor	0.9
Greedy policy epsilon	0.9

Table 1. Hyper-parameters for the DQN.

Order Feature	Description
PickupPostalCode	Pickup location of the order
PickupTime	Pickup time of the order
DeliveryPostalCode	Delivery location of the order
DeliveryTime	Delivery time of the order
Size	Size (weight) of this order
Driver Feature	Description
AddressPostalCode	Residential address of the driver
DeliveryBy	Carrier mode (i.e. van, sedan, motorbike, bicycle)
FullTime	Whether the driver is full time
NoOfDeliveries	Total number of orders were served the driver
CreateOn	First online time of the driver
NoOfJobCancelation	Total number of orders canceled by this driver
CurrentLocation	Current location of the driver

Table 2. Features for xDFM.

Fig. 2. A sample distribution of order locations in a single problem instance.

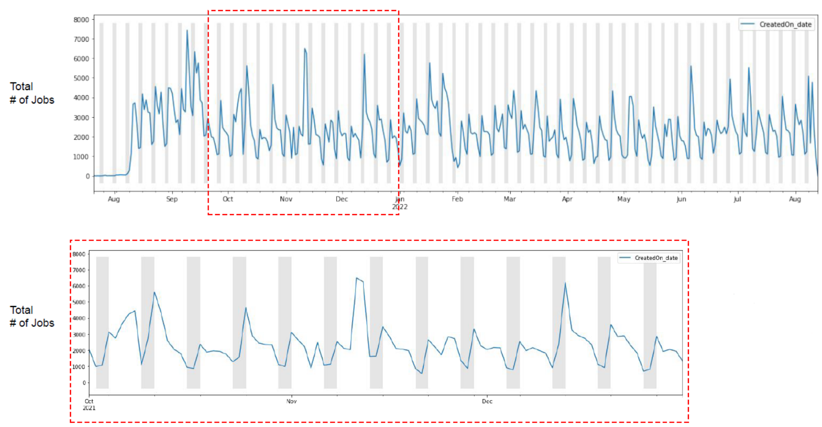

Fig. 3. The distribution of the orders in the data set in terms of date. The x-axis is the dates and the y-axis is the the total number of the orders in one day.