Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)
Special Track on AI for Good

A Prediction-and-Scheduling Framework for Efficient Order Transfer in Logistics

Wenjun Lyu[1,2] , **Haotian Wang**[1] , **Yiwei Song**[1] , **Yunhuai Liu**[3] , **Tian He**[1] , **Desheng Zhang**[2]
[1]JD Logistics,
[2]Rutgers University,
[3]Peking University
wenjun.lyu@rutgers.edu, {wanghaotian18, songyiwei1}@jd.com, yunhuai.liu@pku.edu.cn,
tim.he@jd.com, desheng@cs.rutgers.edu

Abstract

Order Transfer from the transfer center to delivery stations is an essential and expensive part of the logistics service chain. In practice, one vehicle sends transferred orders to multiple delivery stations in one transfer trip to achieve a better trade-off between the transfer cost and time. A key problem is generating the vehicle's route for efficient order transfer, i.e., minimizing the order transfer time. In this paper, we explore fine-grained delivery station features, i.e., downstream couriers' remaining working times in last-mile delivery trips and the transferred order distribution to design a Prediction-and-Scheduling framework for efficient Order Transfer called PSOT, including two components: i) a Courier's Remaining Working Time Prediction component to predict each courier's working time for conducting heterogeneous tasks, i.e., order pickups and deliveries, with a context-aware location embedding and an attention-based neural network; ii) a Vehicle Scheduling component to generate the vehicle's route to served delivery stations with an order-transfer-time-aware heuristic algorithm. The evaluation results with real-world data from one of the largest logistics companies in China show PSOT improves the courier's remaining working time prediction by up to 35.6% and reduces the average order transfer time by up to 51.3% compared to the state-of-the-art methods.

1 Introduction

The popularity of the online-to-offline business has promoted the rapid development of the traditional logistics industry in the past few years [Wang *et al.*, 2018]. Generally, a courier first picks up the order at the source address and takes it to the source delivery station. Then the order is transported through some transfer centers to the destination delivery station. After that, a courier collects the order and delivers it to the destination. In practice, a transfer center is covered by multiple delivery stations in an area, e.g., a district. The order transfer from the transfer center to delivery stations may repeat several times a day if orders arrive at the transfer center in batches. To achieve a better trade-off between the transfer cost and time, a state-of-the-practice solution is that a vehicle sends orders to multiple delivery stations with an order transfer trip. The key problem is to design an efficient vehicle scheduling strategy, i.e., generating the vehicle's route to the served delivery stations to minimize the order transfer time. This problem targets efficient logistics services that impact millions of people's daily lives and economic growth, showing its social importance.

Vehicle scheduling in order transfer is a variant of the traditional traveling salesman problem (TSP), which aims to find a route to all locations with a minimal travel distance. TSP is an NP-hard problem, and some methods have been proposed to solve it in recent years [Bello *et al.*, 2016; Kool *et al.*, 2018]. Different from TSP, which considers the distance between different location pairs, in this paper, we further explore the fine-grained delivery station features to generate the vehicle's route with the goal of minimizing the order transfer time. Specifically, multiple couriers work for a delivery station in practice, and they have different remaining working times for current delivery trips when the order transfer starts, which affects the order transfer time of this delivery station. Besides, the fine-grained transferred order distribution to each courier also affects the order transfer time.

Designing an efficient vehicle scheduling scheme is not trivial due to two challenges: *i) How to capture each courier's remaining working time.* Each courier's remaining working time in a delivery trip should be predicted before getting transferred orders. Due to the distinct amounts and destination address distributions of delivery orders and the real-time pickup orders, the courier takes different working times for different trips; *ii) How to optimize the vehicle scheduling with the goal of minimizing the order transfer time considering complicated factors.* Besides considering the distance between different location pairs in TSP, we also need to focus on the fine-grained delivery station features, i.e., the downstream couriers' remaining working times and the transferred order distribution to each courier, to generate the vehicle's route with the goal of minimizing the order transfer time, which is more complicated.

In this paper, we design a Prediction-and-Scheduling framework to address the above challenges in the order transfer process, which consists of two components: i) a Remaining Working Time Prediction component for predicting each courier's remaining working time of the current delivery trip;

ii) a Vehicle Scheduling component for generating the vehicle's route to all served delivery stations for order transfer. In summary, the key contributions of the paper are as follows:
(i) To the best of our knowledge, we are the first to explore vehicle scheduling for the efficient order transfer process in logistics considering the fine-grained delivery station features, i.e., downstream courier's remaining working time and the transferred order distribution to each courier, which is more challenging compared to traditional TSP.
(ii) We design a Prediction-and-Scheduling framework called PSOT, for efficient order transfer. Specifically, we first predict each courier's remaining working time for the current delivery trip in the corresponding delivery area with a context-aware location embedding and an attention-based neural network when the order transfer process starts. After that, we propose an order-transfer-time-aware heuristic method to generate the vehicle's route by optimizing the order transfer time considering downstream couriers' remaining working times, the transferred order distribution, and the distance between different location pairs.
(iii) We implement PSOT in one of the largest logistics companies in China and conduct experiments based on three-month real-world order data from 20 delivery stations covered by a transfer center involving 391 couriers and 5 million orders. The evaluation results show that PSOT improves the courier's remaining working time prediction and reduces the average order transfer time by up to 35.6% and 51.3%, respectively, compared to state-of-the-art methods.

2 Motivation

2.1 Background

The logistics network shows the hierarchical topology structure, that is, a transfer center serves multiple delivery stations, and a delivery station provides the last-mile delivery services to multiple delivery areas, each of which is covered by a courier. The order transfer and last-mile delivery processes are shown in Fig. 1. A set of orders are transferred from the transfer center to served delivery stations sequentially by a vehicle. After the transferred orders arrive at a delivery station, each courier collects a sub-set of orders destined for one delivery area and delivers them to customers consecutively.

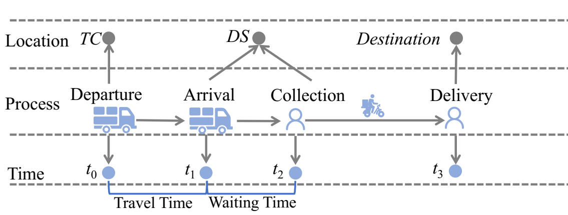


Figure 1: Order Transfer and Last-mile Delivery Process

Definition 1 (Last-mile Delivery Trip). The process between the courier departing from and back to the delivery station is a last-mile delivery trip. During the last-mile delivery trip, the courier has to deliver orders collected at the delivery station to customers. The courier also needs to finish the real-time order pickup tasks with strict promised service times. Specifically, after a customer places an order pickup request at the logistics platform and the courier gets the assigned task, he/she has to pick up the order within a given time, e.g., one hour, to ensure the customer's experience with the logistics service.

Order transfer and last-mile delivery processes are generally conducted several times a day. For example, the orders are transported from the transfer center to all served delivery stations in two batches, i.e., the morning and afternoon batches. As a result, to provide timely order delivery, each courier needs to conduct two last-mile delivery trips for two batches of orders. During the order transfer process of the first batch of orders, each courier is with the same status, i.e., stays at the delivery station and without the assigned task. In this paper, we mainly focus on the order transfer process of the non-first batch of orders, where couriers have been assigned the former batch of delivery tasks and started the last-mile delivery trips with different remaining working times when we need to generate the vehicle's route for order transfer of the current batch of orders.

Definition 2 (Remaining Working Time). The courier's remaining working time is the time for conducting remaining heterogeneous tasks, i.e., order pickups and deliveries, and back to the delivery station, which is real-time updated with the courier's status.

Definition 3 (Order Transfer Time (OTT)). As shown in Fig. 1, given the order departure time from the transfer center t_0, the order arrival time to the delivery station t_1, and the order collection time by the courier t_2. Order transfer time is defined as $t_2 - t_0$, which includes the travel time $t_1 - t_0$, and the waiting time $t_2 - t_1$. If the courier has finished the former last-mile delivery trip and is waiting for the collection of newly transferred orders at the delivery station before t_1, which means that the courier can collect and deliver orders destined for his/her delivery area when the transferred orders arrive right away, then the order waiting time is 0. Otherwise, the order waiting time equals the courier's back time to the delivery station minus t_1.

2.2 Problem Formulation

In this work, we focus on the vehicle's route generation problem in the order transfer process, which includes the following two tasks: (i) The first task of this work is to predict each courier's remaining working time in a last-mile delivery trip when the order transfer starts, given the destination distribution of delivery orders and the generation and the promised service times of the pickup orders. Based on the predicted courier's remaining working time and the current time, we get the courier's back time to the delivery station, which is then used as the input for vehicle scheduling; (ii) The second task of this work is to design an effective vehicle scheduling scheme in the order transfer process with the goal of minimizing the average OTT (AOTT) for all transferred orders considering the downstream couriers' remaining working times, the transferred order distribution to each courier, and the distance between different delivery station pairs.

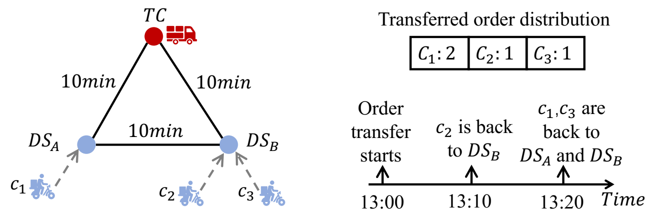


Figure 2: An Example of the Impact of Couriers' Remaining Working Times and the Transferred Order Distribution on the Order Transfer Time

2.3 Observations

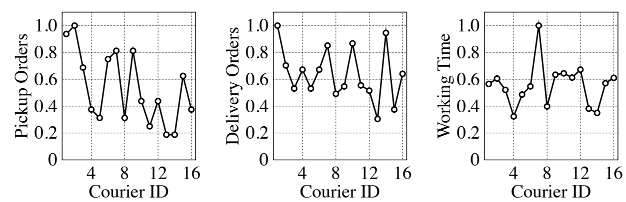
Opportunity

As shown in Fig. 1, the order waiting time at the delivery station is a part of the order transfer time, which is correlated with the downstream couriers' remaining working times and the transferred order distribution of the last-mile delivery trips. Fig. 2 shows an example of how couriers' remaining working times and the transferred order distribution affect the order waiting time and the order transfer time. One courier c_1 serves the delivery station DS_A, and two couriers c_2 and c_3 serve DS_B. Transferred orders to c_1, c_2, and c_3 are 2, 1, and 1, respectively. The vehicle travel time between arbitrary two locations of TC, DS_A, and DS_B is set as 10 min. Assume that when the order transfer process starts at $13:00$, c_1, c_2, and c_3 are back to the delivery station at $13:20$, $13:10$, and $13:20$, respectively, based on the predicted remaining working times.

There are two kinds of vehicle routes for order transfer: i) If the vehicle's route is $TC \rightarrow DS_A \rightarrow DS_B$, the total travel time for orders to DS_A is 20 min (2×10 min). c_1 is back to DS_A 10 min later than the orders' arrival, so the total waiting time for orders to DS_A is 20 min. For orders to DS_B, the total travel time and waiting time are 40 min (2×20 min) and 0 min, respectively. As a result, for transferred orders, the total waiting time and order transfer time are 20 min and 80 min; ii) If the vehicle's route is $TC \rightarrow DS_B \rightarrow DS_A$, the total travel time and waiting time for orders to DS_B are 20 min and 10 min, respectively. The total travel time and waiting time for orders to DS_A are 40 min and 0 min, respectively. As a result, for transferred orders, the total waiting time and order transfer time are 10 min and 70 min, respectively. The route $TC \rightarrow DS_B$ and DS_A is better because of the smaller order transfer time. The waiting time for orders to DS_A is the same with two routes, but the value is different for orders to DS_B because of the different courier's remaining working time and back time to the delivery station. It is necessary and effective to utilize the couriers' remaining working times and the transferred order distribution as the input information to design the vehicle scheduling algorithm for order transfer to minimize the order transfer time.

Challenges

The first challenge of generating the vehicle's route is to capture each courier's remaining working time for the current last-mile delivery trip when the order transfer process starts. In practice, due to the different amount and destination distribution of the delivered orders and the uncertainty of the pickup requests' generation during a last-mile delivery trip, couriers need to take different times to finish the tasks and back to the delivery station. We analyze the order data from a delivery station, and the results in Fig. 3 show that pickup and delivery orders are significantly different for different couriers' last-mile delivery trips, which causes the varying working times. Besides, the correlation between the courier's working time and the number of heterogeneous tasks, i.e., pickup and delivery orders, are complicated and can't be captured by a simple linear regression model. The second challenge is to generate the vehicle's route with the goal of minimizing the order transfer time, which is an NP-hard problem. We need to take both the couriers' remaining working time prediction results, the distribution of the transferred orders, and the distance between different delivery station pairs to generate an efficient vehicle route for the order transfer process.



(a) Pick up Orders (b) Delivery Orders (c) Working Time

Figure 3: Heterogeneous Order Distribution and Courier Working Time Analysis

3 Design

3.1 Design Overview

The Prediction-and-Scheduling framework `PSOT` is shown in Fig. 4, including two main components:

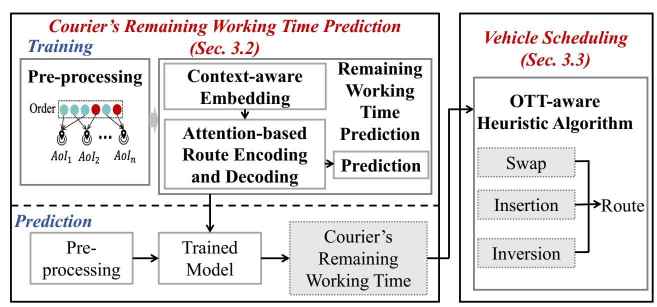


Figure 4: Framework of `PSOT`

(i) Courier's Remaining Working Time Prediction. This component aims to predict the courier's remaining working time in a last-mile delivery trip when the order transfer starts. The locations are first embedded through a context-aware embedding layer. Then an attention-based route encoding and decoding and a prediction part are designed to get the remaining working time. The output is utilized to get the courier's back time to the delivery station.

(ii) Vehicle Scheduling. This component aims to generate the vehicle's route to served delivery stations by minimizing the

order transfer time with an OTT-aware heuristic algorithm considering downstream couriers' remaining working times and the transferred order distribution.

3.2 Courier's Remaining Working Time Prediction

Pre-processing

Each order delivery or pickup task has a service address, which can be represented as a GPS point, including the latitude and the longitude. We divide the delivery area into different area-of-interests (AoIs) and map each order in an AoI, e.g., a community, based on comparing the order's GPS point and the AoIs' boundaries. compared to the grid-based partition [Liang *et al.*, 2019], the AoI-based partition can capture the semantic information [Wu and Wu, 2019]. Each AoI is regarded as one location, which is the finest granularity used in this work. Generally, the courier finishes tasks, e.g., the pickup and delivery orders, in a location sequentially and then continue the trip to other locations.

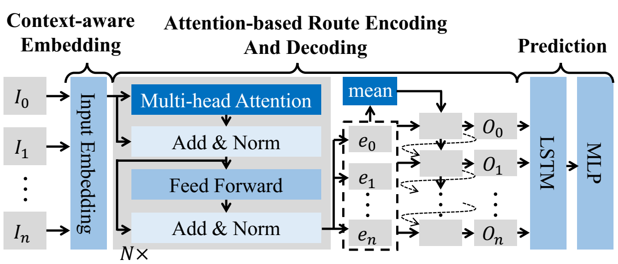


Figure 5: Remaining Working Time Prediction

Features for Representing a Location

In this work, we extract the following four features from heterogeneous orders and locations to represent a location: *(i) Number of Delivery Orders.* A location may have some delivery orders, affecting the courier's route and working time. *(ii) Number of Pickup Orders.* Similar to the above feature, a location may also have some order pickup requests. *(iii) Earliest Promised Service Time.* Each order pickup request is with a promised service time. We choose the earliest promised service time for pickup orders at a location as one feature for prediction. *(iv) Distance to the Current Location.* The courier's route is correlated to the courier's current location, which affects the prediction of the remaining working time.

Remaining Working Time Prediction Model

The courier's remaining working time prediction model is shown in Fig. 5, which includes three parts: i) a context-aware embedding to represent locations; ii) an attention-based route encoding and decoding to get the vehicle's route for the order transfer process; iii) a Long Short-Term Memory (LSTM) layer and a Feed Forward (FF) layer to output the predicted courier's working time.

The features of locations are taken as the input to a feed-forward layer to get the location representations:

$$h_i^0 = W^0 x_i + b^0 \tag{1}$$

where x_i is the 4-dimensional input features, and h_i^0 is the node embedding.

The attention-based route encoding includes N transformer blocks with the same structure. Each block includes two sub-layers, i.e., a Multi-Head Attention (MHA) layer and a FF layer. The message-passing process is formulated as:

$$\hat{h}_i = BN^l(h_i^{l-1} + MHA_i^l(h_1^{l-1}, ..., h_n^{l-1})) \tag{2}$$

$$h_i^l = BN^l(\hat{h}_i + FF^l(\hat{h}_i)) \tag{3}$$

Specifically, we first calculate the single-head attention as

$$\begin{aligned} head_i &= Attention(W^Q h_i, W^K h_i, W^V h_i) \\ &= softmax(\frac{W^Q h_i (W^K h_i)^T}{\sqrt{d_k}}) W^V h_i \end{aligned} \tag{4}$$

where W^Q, W^K, and W^V are the parameters for the query, key, and value, respectively. d_k is the dimension of the key and query vector. After that, the multi-head attention value for a node i is calculated as

$$MHA_i(h_1, h_2, ..., h_n) = \sum_{m=1}^{M} head_m W^O \tag{5}$$

where $head_m$ is the single-head attention. W^O is the parameter matrix. The MHA layer executes $head_m$ in parallel with the same input vector and then combines all the output vectors from all single-head attention to generate the final output.

In the FF sub-layer and BN, we have

$$FF(\hat{h}_i) = W_{ff}^1 \times ReLu(W_{ff}^0 \times \hat{h}_i + b_{ff}^0) + b_{ff}^1 \tag{6}$$

$$BN(h_i) = W_{bn} \odot \overline{BN}(h_i) + b_{bn} \tag{7}$$

where W_{ff}^0, W_{ff}^1, b_{ff}^0, b_{ff}^1, W_{bn} and b_{bn} are parameters. $ReLu$ is the ReLu activation. $\odot$ is the element-wise product and $\overline{BN}$ represents the batch normalization.

After getting the embedding of different locations h^l, we further adopt attention-based decoding to get the vehicle's route of the order transfer process. Specifically, at each time step t, the decoder outputs a location based on the encoding embeddings and the output generated before t. We utilize the similar decoding structure presented in [Kool *et al.*, 2018], which iteratively outputs the probabilities of the unvisited locations and selects the location with the largest probability as the next visiting location to generate the route. We add the LSTM and FF layers to get the courier's remaining working time with the route output by the attention-based route encoding and decoding. The loss function for training is

$$l = \frac{1}{n} \sum_{i=1}^{n} (y_i - y_i^{pred})^2 \tag{8}$$

where y_i and y_i^{pred} are the actual and predicted courier's remaining working time, respectively. n is the number of training samples. Given the predicted courier's remaining working time and the start time of the order transfer process, we can get the courier's back time to the delivery station, which is further utilized for the vehicle scheduling algorithm design in the order transfer process.

3.3 Vehicle Scheduling

In the Vehicle Scheduling component, we generate the vehicle's route to served delivery stations for order transfer. Specifically, we first initialize the vehicle's route and then utilize an OTT-aware heuristic algorithm to improve the route, i.e., reduce the Average OTT for transferred orders. We choose a distance-based greedy algorithm to initialize the vehicle's route. Specifically, the transfer center is selected as the start location, and we choose the nearest delivery station as the next visit location iteratively to get the initialized route.

Based on the generated route, we calculate the average order transfer time. Specifically, given the vehicle's departure time from the transfer center t^{start} and the vehicle's route $route$, we first calculate the vehicle's arrival time to a delivery station DS_i as t_i^{arr} based on the vehicle's speed and the distance between different pair of locations, i.e., delivery stations and the transfer center. Assume there are j couriers working for the delivery station DS_i. With each courier's remaining working time prediction as described in Section 3.2, we can get all j couriers back time to DS_i as $[t_i^1, t_i^2, ..., t_i^j]$. The number of transferred orders to j couriers is represented as $[n_i^1, n_i^2, ..., n_i^j]$. As a result, the total travel time for orders to DS_i is represented as:

$$t_i^{travel} = n_i \times (t_i^{arr} - t^{start}), n_i = \sum_{k=1}^{j} n_i^k \tag{9}$$

The total waiting time for orders to DS_i is represented as:

$$t_i^{waiting} = \sum_{k=1}^{j} max\{0, t_i^k - t_i^{arr}\} \times n_i^k \tag{10}$$

As a result, the total transfer time for orders to DS_i is $t_i^{travel} + t_i^{waiting}$. Assume there are N delivery stations and the total amount of transferred orders is M, then the average order transfer time is calculated as:

$$AOTT^{route} = \frac{\sum_{i=1}^{N}(t_i^{travel} + t_i^{waiting})}{M} \tag{11}$$

The main idea of the OTT-aware heuristic algorithm is to conduct mutation operations to improve the vehicle's route with a smaller average order transfer time by considering the downstream couriers' remaining working times and the transferred order distribution. The mutation operations in the OTT-aware heuristic algorithm include the following three categories: **(i) Swap**: Randomly exchanges the positions of two delivery stations in the vehicle's route; **(ii) Insertion**: Randomly extracts one delivery station from the vehicle's route and re-inserts it into the random position in the route; **(iii) Inversion**: Randomly selects the start and end delivery stations in the route to get a sequence and then reverses this sequence in the route. The detailed operation process of the OTT-aware heuristic algorithm is shown in Algorithm 1. If AOTT is reduced after a mutation operation, the operation is successful and we adopt the new route. Otherwise, the operation is failed. The algorithm stops when the number of consecutive failed mutation operations num equals the threshold S.

Algorithm 1: OTT-aware Heuristic Algorithm

Input:
$route$: The initialized vehicle's route;
S: The threshold for stopping the algorithm;
$AOTT^{route}$: Average OTT of transferred orders with route $route$;
Output:
$route$: The optimized relay route;

$num \leftarrow 0$;
while $num < S$ **do**
  Randomly select one from three mutation operations;
  Mutate the vehicle's current route $route$ to generate new route new_route;
  Calculate the average OTT with new_route as $AOTT^{new\_route}$;
  if $AOTT^{new\_route} < AOTT^{route}$ **then**
    $AOTT^{route} \leftarrow AOTT^{new\_route}$;
    $route \leftarrow new\_route$;
    $num \leftarrow 0$;
  else
    $num \leftarrow num + 1$;

4 Evaluation

4.1 Methodology

Data. The data used for evaluation include order data, couriers' reporting data, and couriers' trajectory data collected from one of the largest logistics companies in China, involving 20 delivery stations covered by one transfer center, 391 couriers, and 5 million orders in three months.

Implementation. The courier's remaining working time prediction is based on the morning batch of orders' delivery. We utilize two-month data as the training data, and the remaining data are used for testing the model's performance. The dimension of the node embedding is 64. The FF layer includes one layer with 256 dimensions and ReLu activation, and the MHA layer includes 8 heads. Note that couriers don't need to report their back time to the delivery station. As a result, we get the ground truth data, i.e., the couriers' remaining working times, from their trajectory data with the stay-point detection [Li *et al.*, 2008; Ruan *et al.*, 2020] and the comparison of the GPS coordinates, i.e., latitudes and longitudes, of delivery stations. The afternoon batch of orders is regarded as the transferred orders, which are transported by the vehicle traveling from the transfer center to all served delivery stations sequentially based on the generated vehicle route.

Metrics. For the courier's remaining working time prediction, we use Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) as metrics. For vehicle scheduling, our goal is to minimize the average order transfer time of the transferred orders, which can be used as the metric to show the effectiveness of `PSOT`. We also show the vehicle's total travel distance as a metric to show the cost of the vehicle scheduling for order transfer. Given a route $route$, the total travel distance is the sum of the distance between any two adjacent nodes in $route$. Note that the first node is the transfer center, and the last node is the final visited delivery station by the vehicle before returning back to the transfer center.

Baselines. For the courier's remaining working time prediction, we choose the following three methods as the baselines: **(i) MLP**: This method considers the number of orders at each location as the input to get the remaining working time with a multi-layer neural network. **(ii) Distance-aware method+SB-TTE** [Wang *et al.*, 2019] **(DS)**: This method first utilizes the distance-based greedy algorithm, i.e., selecting the nearest delivery station as the next location iteratively to generate the route. Then it predicts the travel time between any adjacent two delivery stations in the route based on a historically similar trip. To improve the prediction results, we add the courier's stay time prediction at a location considering the number of orders with MLP. **(iii) OSquare Route** [Zhang *et al.*, 2019] + **SB-TTE** [Wang *et al.*, 2019] **(OS)**: This method first utilizes the XGBoost model to predict the courier's route and then predicts the remaining working time with the same way as the above baseline. For vehicle scheduling, we choose the following three methods as the baselines for comparison: **(i) Amount-based Greedy Algorithm (AGA)**: The transfer center is selected as the start location, and the delivery station with the most amount of transferred orders is selected as the next location iteratively to generate the vehicle's route. **(ii) Distance-based Greedy Algorithm (DGA)** [Nilsson, 2003]: The transfer center is selected as the start location, and the nearest delivery station is selected as the next location iteratively to generate the vehicle's route. **(iii) DGA+Distance-based Mutation Algorithm (DGMA)**: This method first initializes the vehicle's route with DGA, and then the vehicle route is improved by optimizing the vehicle's travel distance with the mutation operations and threshold similar to PSOT.

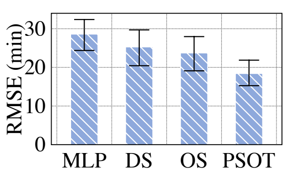


Figure 6: RMSE of the Courier's Remaining Working Time Prediction

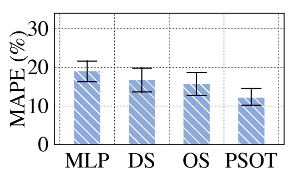


Figure 7: MAPE of the Courier's Remaining Working Time Prediction

4.2 The Performance of the Courier's Remaining Working Time Prediction

We first show the performance of the courier's remaining working time prediction in PSOT. As shown in Fig. 6 and Fig. 7, MLP achieves the worst performance compared to other baselines. OS performs better than DS because the courier generally has his/her own preference for the visit sequence to different locations in the corresponding delivery area. PSOT performs best compared to other baselines because the context information of each location is considered in an end-to-end learning manner. Specifically, PSOT improves RMSE by 22.3% and 35.6% compared to MLP and OS, respectively.

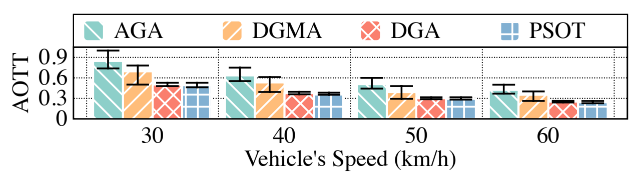
Methods	Number of Delivery Stations			
	15		20	
	AOTT (%)	TD (%)	AOTT (%)	TD (%)
AGA	74.97	79.45	100	100
DGA	44.38	38.77	51.47	42.03
DGMA	62.28	**36.34**	53.62	**41.74**
PSOT	**43.03**	39.03	**48.70**	42.14

* TD means travel distance. The values of AOTT and the travel distance are normalized based on the largest value

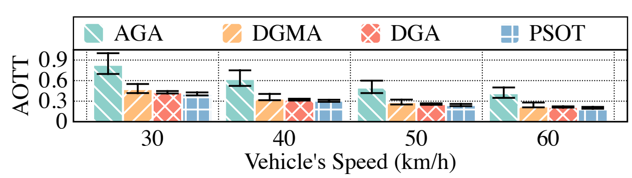
Table 1: Overall Performance with Different Number of Delivery Stations

4.3 The Performance of the Vehicle Scheduling

Overall Performance. We first show the overall performance of different vehicle scheduling schemes. The default value of the vehicle's speed and the threshold S in the OTT-aware heuristic algorithm are set as 60 km/h and 300, respectively. The start time of the order transfer process is set as 13:30:00. As shown in Table 1, AGA achieves the worst performance, i.e., the largest AOTT, because it is a time-agnostic method and always serves the delivery station with the most number of transferred orders. DGA and DGMA try to find a route with the shortest total travel distance for the order transfer process. A route with a shorter travel distance produces shorter travel time, which generally may reduce AOTT. PSOT achieves the smallest AOTT compared to other vehicle scheduling schemes because it considers AOTT, the transferred order distribution, and the distance with an OTT-aware heuristic algorithm. We also show the cost of different methods for order transfer. compared to DGMA, PSOT doesn't increase much vehicle's travel distance as shown in Table 1.



(a) 15 Delivery Stations

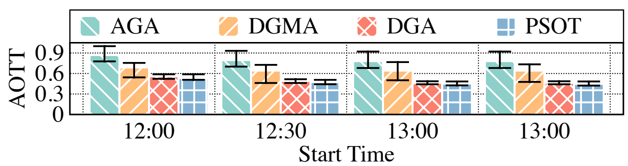


(b) 20 Delivery Stations

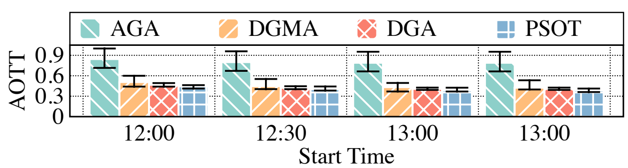
Figure 8: The Impact of the Vehicle's Speed

The Impact of the Vehicle's Speed. The vehicle's speed affects the vehicle's arrival time at delivery stations, which further affects the order transfer time because the order waiting time at the delivery station is affected. As a result, we show the impact of the vehicle's speed on AOTT. As shown in Fig. 8, the larger vehicle's speed, the smaller AOTT because of the smaller vehicle's travel time for order transfer. `PSOT` achieves the best performance with different vehicle speeds compared to other methods. Specifically, `PSOT` reduces AOTT by 50.9%∼51.3% and 9%∼14.5% compared to AGA and DGMA, respectively, with 20 delivery stations.

The Impact of the Order Transfer Start Time. The start time of the order transfer process from the transfer center affects the order transfer time. As shown in Fig. 9, the later the start time, the smaller AOTT because more couriers have finished their last-mile delivery trips and waiting for the transferred orders at delivery stations, which reduces the waiting time of transferred orders at delivery stations. `PSOT` achieves the best performance with varying order transfer start times. Specifically, `PSOT` reduces AOTT by 36.5%∼42.6% and 47.6%∼51.3% compared to AGA with 15 and 20 delivery stations, respectively.



(a) 15 Delivery Stations



(b) 20 Delivery Stations

Figure 9: The Impact of the Order Transfer Start Time

The Impact of the Threshold S. We evaluate the impact of the threshold S of the OTT-aware heuristic algorithm on AOTT. As shown in Fig. 10, the larger value of S, the smaller AOTT because more mutation operations are conducted to generate a better vehicle's route.

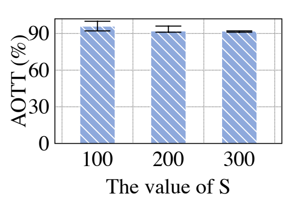


Figure 10: The Impact of S

5 Related Work

5.1 Order Transportation in Logistics

Order transportation in logistics has become a hot research topic in recent years with the popularity of the online-to-offline business. Gao et al. [Gao *et al.*, 2022] leverage public buses to relay UAVs for on-demand delivery with the stochastic mission time and location. Zong et al. [Zong *et al.*, 2022] solve the large-scale vehicle routing problem hierarchically in the logistics delivery systems with reinforcement learning. Li et al. [Li *et al.*, 2022] design a time-aware batch-matching algorithm with theoretical matching approximation bound analysis for efficient courier-task matching for real-time city logistics systems. Ma et al. [Ma *et al.*, 2021] propose a hierarchical reinforcement learning framework to optimize the order pickup and delivery process in the logistics domain. Some works focus on the collaboration among multiple logistics companies, e.g., the delivery station sharing [Zibaei *et al.*, 2016; Zhang *et al.*, 2022] for improving the delivery efficiency, and the delivery area sharing [Ko *et al.*, 2020] for reducing the delivery cost.

5.2 Courier Mobility Prediction

Couriers' mobility prediction has been studied in recent works. Zhang et al. [Zhang *et al.*, 2019] predict the courier's route for picking up and delivering orders in instant delivery with an XGBoost-based model. Wen et al. [Wen *et al.*, 2022] predict the order pickup and delivery route with a dynamic spatial-temporal graph neural network. Ruan et al. [Ruan *et al.*, 2022] predict the courier's stay time at a location for delivering orders in last-mile logistics delivery via a spatial metal-learning-based neural network. Wu et al. [Wu and Wu, 2019] predict the order delivery time in the logistics delivery system with a spatial-temporal sequential neural network model. Wu et al. [Wu and Wu, 2019] predict the courier's arrival time to each location for last-mile order delivery with a spatial-temporal sequential neural network.

6 Conclusion

In this paper, we design a Prediction-and-Scheduling framework `PSOT` for efficient order transfer from the transfer center to multiple delivery stations including: i) a Courier's Remaining Working Time Prediction component to predict each courier's remaining working time for conducting heterogeneous tasks when the order transfer starts with context-aware location embedding and an attention-based neural network; and ii) a Vehicle Scheduling component to generate the vehicle's route to served delivery stations with an OTT-aware heuristic method by minimizing the average order transfer time considering downstream couriers' remaining working times, the transferred order distribution, and the distance between different location pairs. The evaluation results show that `PSOT` improves the courier's remaining working time prediction and reduces the average order transfer time by up to 35.6% and 51.3%, respectively, compared to state-of-the-art methods.

Acknowledgements

The work is in part by the National Natural Science Foundation of China (NSFC) under Grant 61925202 and 2030 major projects of scientific and technological innovation under Grant No. 2021ZD0114200.

References

[Bello *et al.*, 2016] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

[Gao *et al.*, 2022] Junhui Gao, Yan Pan, Zhigang Li, Qingye Han, and Qianwu Chen. Leveraging public buses to relay uavs for on-demand applications. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 907–909, 2022.

[Ko *et al.*, 2020] Seung Yoon Ko, Ratna Permata Sari, Muzaffar Makhmudov, and Chang Seong Ko. Collaboration model for service clustering in last-mile delivery. *Sustainability*, 12(14):5844, 2020.

[Kool *et al.*, 2018] Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.

[Li *et al.*, 2008] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, pages 1–10, 2008.

[Li *et al.*, 2022] Yafei Li, Qingshun Wu, Xin Huang, Jianliang Xu, Wanru Gao, and Mingliang Xu. Efficient adaptive matching for real-time city express delivery. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Liang *et al.*, 2019] Yuxuan Liang, Kun Ouyang, Lin Jing, Sijie Ruan, Ye Liu, Junbo Zhang, David S Rosenblum, and Yu Zheng. Urbanfm: Inferring fine-grained urban flows. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3132–3142, 2019.

[Ma *et al.*, 2021] Yi Ma, Xiaotian Hao, Jianye Hao, Jiawen Lu, Xing Liu, Tong Xialiang, Mingxuan Yuan, Zhigang Li, Jie Tang, and Zhaopeng Meng. A hierarchical reinforcement learning based optimization framework for large-scale dynamic pickup and delivery problems. *Advances in Neural Information Processing Systems*, 34:23609–23620, 2021.

[Nilsson, 2003] Christian Nilsson. Heuristics for the traveling salesman problem. *Linkoping University*, 38:00085–9, 2003.

[Ruan *et al.*, 2020] Sijie Ruan, Zi Xiong, Cheng Long, Yiheng Chen, Jie Bao, Tianfu He, Ruiyuan Li, Shengnan Wu, Zhongyuan Jiang, and Yu Zheng. Doing in one go: Delivery time inference based on couriers' trajectories. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2813–2821, 2020.

[Ruan *et al.*, 2022] Sijie Ruan, Cheng Long, Zhipeng Ma, Jie Bao, Tianfu He, Ruiyuan Li, Yiheng Chen, Shengnan Wu, and Yu Zheng. Service time prediction for delivery tasks via spatial meta-learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3829–3837, 2022.

[Wang *et al.*, 2018] Fangxin Wang, Yifei Zhu, Feng Wang, and Jiangchuan Liu. Ridesharing as a service: Exploring crowdsourced connected vehicle information for intelligent package delivery. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE, 2018.

[Wang *et al.*, 2019] Hongjian Wang, Xianfeng Tang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. A simple baseline for travel time estimation using large-scale trip data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–22, 2019.

[Wen *et al.*, 2022] Haomin Wen, Youfang Lin, Xiaowei Mao, Fan Wu, Yiji Zhao, Haochen Wang, Jianbin Zheng, Lixia Wu, Haoyuan Hu, and Huaiyu Wan. Graph2route: A dynamic spatial-temporal graph neural network for pick-up and delivery route prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4143–4152, 2022.

[Wu and Wu, 2019] Fan Wu and Lixia Wu. Deepeta: a spatial-temporal sequential neural network model for estimating time of arrival in package delivery system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 774–781, 2019.

[Zhang *et al.*, 2019] Yan Zhang, Yunhuai Liu, Genjian Li, Yi Ding, Ning Chen, Hao Zhang, Tian He, and Desheng Zhang. Route prediction for instant delivery. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–25, 2019.

[Zhang *et al.*, 2022] Keyang Zhang, Jose Javier Escribano Macias, Dario Paccagnan, and Panagiotis Angeloudis. The competition and inefficiency in urban road last-mile delivery. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 1473–1481, 2022.

[Zibaei *et al.*, 2016] Sedighe Zibaei, Ashkan Hafezalkotob, and Seyed Sajad Ghashami. Cooperative vehicle routing problem: an opportunity for cost saving. *Journal of Industrial Engineering International*, 12(3):271–286, 2016.

[Zong *et al.*, 2022] Zefang Zong, Hansen Wang, Jingwei Wang, Meng Zheng, and Yong Li. Rbg: Hierarchically solving large-scale routing problems in logistic systems via reinforcement learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4648–4658, 2022.