A Personalized Service for Scheduling Express Delivery Using Courier Trajectories

Yanchun Sun, Kui Wei, Zijian Qiao, Jiu Wen, Tianyuan Jiang
Institute of Software, School of Electronics Engineering and Computer Science, Peking University
Key Laboratory of High Confidence Software Technologies, Ministry of Education Beijing 100871, China
{sunyc, ccweikui, qiaozijian, wenjiu, jtyuan}@pku.edu.cn

Abstract*—With the increasing demand for express delivery, a courier needs to deliver many tasks in one day and it's necessary to deliver punctually as the customers expect. At the same time, they want to schedule the delivery tasks to minimize the total time of a courier's one-day delivery, considering the total travel time. However, most of scheduling researches on express delivery focus on inter-city transportation, and they are not suitable for the express delivery to customers in the "last mile". To solve the issue above, this paper proposes a personalized service for scheduling express delivery, which not only satisfies all the customers' appointment time but also makes the total time minimized. In this service, personalized and accurate travel time estimation is important to guarantee delivery punctuality when delivering shipments. Therefore, the personalized scheduling service is designed to consist of two basic services: 1) personalized travel time estimation service for any path in express delivery using courier trajectories, 2) an express delivery scheduling service considering multiple factors, including customers' appointments, one-day delivery costs, etc., which is based on the accurate travel time estimation provided by the first service. We evaluate our proposed service based on extensive experiments, using GPS trajectories generated by more than 1000 couriers over a period of two months in Beijing. The results demonstrate the effectiveness and efficiency of our method.

Keywords- personalized service; express delivery scheduling; travel time estimation; courier trajectories

I. INTRODUCTION

With the continuous growing of e-business, the logistics industry is developing rapidly. There is an increasing requirement for punctuality and efficiency of express delivery service. In light of this requirement, a great deal of related researches have been conducted. However, most of the literatures focus on the scheduling of inter-city transportation, rather than the "last mile" delivery [1], which is of most concern to the customers. Because all customers may have different time schedules, it is especially important for an express company to deliver goods punctually as customers expect. At the same time, the express companies certainly hope to minimize the total travel time for a courier in one-day delivery. Therefore, a courier is facing a tough problem that he needs to design the task schedule reasonably at the beginning of a day's work, which should not only satisfy all the customers' appointment time but also make the total time minimized. In this paper, we propose a personalized service for scheduling express delivery using courier trajectories to solve the aforesaid problem. In this service, we design two basic services: a) personalized travel time estimation service for any path in express delivery using courier trajectories, b) an express delivery scheduling service considering multiple factors, including customers' appointment, one-day delivery costs, etc., which is based on the accurate travel time estimation in the first service.

Personalized travel time estimation in express delivery becomes an important task with the development of express delivery industry. In the first service of the personalized service, we propose a link-based approach based on the travel speed estimation on road segments using historical courier trajectories. Firstly, there are some particularities for travel time estimation based on courier trajectories. For example, there are many different types of vehicles being used to deliver shipments in China, such as bicycles, electric bicycles, motorcycles, and so on. What's more, there are some sources of bias for courier trajectory such as incomplete coverage of route. For example, as Figure 1 shows, the path from point S to E consists of two incomplete road segments as road A and road C. It's not accurate to estimate the travel time of the path using the complete travel time of the road A and road C. To address the vehicles diversity and the bias, we use the travel speed and travel length instead of the complete estimated travel time on the road segments to calculate the total route travel time. Secondly, there are different travel speeds on the road segments in different time slots because the travel speed will slow down at busy time. Therefore, we model different drivers' travel speeds on different road segments in different time slots with a three-dimensional tensor. What's more, we construct a time matrix to store the intersection delay time between different road segments. Thirdly, because many road segments may not be traveled by any GPS-equipped vehicle in historical trajectories, there are many missing values in the three-dimensional speed tensor. To address the data sparsity problem, we apply non-negative tensor factorization to fill in the speed tensor's missing value because there are some latent features underlying the interactions among users, time slots and road segments in the speed tensor. Therefore, given any path represented as a sequence of connected road segments, we can estimate the partial travel time on each road segment using the travel length and estimated travel speed, then we can estimate the path travel time by concatenating each road segment's travel time and the intersection delay time.

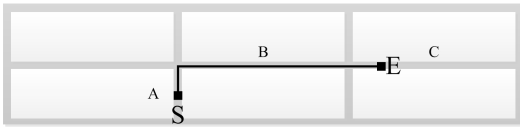

Figure 1. Examples of express delivering vehicles route

In the second service of the personalized service, we propose an approach to solve the "last mile" delivery problem mentioned above. It consists of two essential parts. In the first part, aiming at satisfying the customers' appointment time and minimalizing one-day delivery cost, we design a personalized express delivery scheduling model based on the personalized travel time estimated by the first service for every courier. The model also considers many real-life factors, such as the period for a customer to sign the delivered goods, the V.I.P. level of the customer, etc. Therefore, it can provide each courier with a personalized and reasonable schedule of his one-day delivery tasks, in order to improve delivery efficiency and customers' satisfaction. However, during scheduling process, there may be some objective conflicts among the delivery tasks. For example, there are two delivery tasks A and B which are both appointed to be delivered between 9:00 a.m. and 9:30 a.m., meanwhile, it takes at least 40 minutes to travel from one point to another. In this case, we can find these two tasks cannot be satisfied simultaneously. Therefore, in the second part, we propose another model to deal with the conflicted delivery tasks. The model will notify the conflicted customers to make new appointments, which are recommended by analyzing the task information and the current schedule. Therefore, given one-day delivery tasks of one courier, the service can not only provide the courier with a personalized and rational schedule, but also recommend an available appointments list to the customers with conflicted task.

To summarize, the major contributions of this paper lie in the following aspects:

1) We present a new approach to estimating the personalized travel time in express delivery using historical courier trajectories.
2) We propose an innovative method to schedule express delivery, which not only satisfies all the customers' appointment time but also makes the total cost minimized.
3) We develop a personalized scheduling service, which combines the accurate travel time estimation and delivery task scheduling to improve the efficiency and customers' satisfaction in express delivery.

The rest of the paper is organized as follows. Section II introduces the related work. Section III details the personalized travel estimation service. Then Section IV details express delivery scheduling service. Experimental observations are discussed in Section V. Finally, section VI presents concluding remarks and the future work.

II. Related Work

This section covers existing literature in travel time estimation and express delivery scheduling.

A. Travel Time Estimation

Many research studies have investigated in travel time estimation for cars as reviewed in [2]. Related works in the domain of travel time estimation mainly focus on two methods: road segment-based travel time estimation and path-based travel time estimation [3]. In our paper, we apply the road segment-based travel time estimation, which follows the idea of estimating the travel time of individual road segments and then summing up the travel times of the road segments belonging to one path. De Fabritiis C et al. present the method to predict the travel speed of a road segment using trajectories collected by cars equipped with a specific device [4]. Jenelius E et al. present a statistical network model to separate trip travel times into link travel times and intersection delays and allow the correlation between travel times on different network links based on a spatial moving average (SMA) structure [5]. Different from these works using the car trajectories collected by specific device, we propose a new approach that focuses on dealing with the sparse courier trajectories collected by mobile phones, which applies non-negative tensor factorization to deal with the data sparsity.

B. Express Delivery Scheduling

Current researches on express delivery mainly focus on transportation between distribution centers rather than terminal users and the organization structure of distribution centers [6,7]. Marius M. Solomon reduce the express delivery scheduling problem to the Vehicle Routing Problem with Time Windows (VRPTW) [8], which is an important generalization of the original Vehicle Routing Problem (VRP) [9]. In VRP, a set of clients with known demands must be served exactly once by vehicles of limited capacity with minimized total cost. The VRPTW adds time constraints to the VRP, so that each client has a time window within which he must be served [9].

Latest researches on VRPTW algorithms have paid a lot of attention to metaheuristic methods: Simulated annealing algorithm [10], Tabu search [11], Genetic algorithm [12], Ant-Colony Optimization (ACO) algorithm [13,14]. Considering the unpredictability of the "last-mile" delivery process, ACO is chosen for our service from all the above algorithms, because it can be run continuously and adapt to changes in real time [15].

By analyzing the related work above, we find out that the traditional VRPTW methods have some problems when applied to the "last-mile" delivery. One major problem is that the time conflict problem can no longer be solved by simply increasing the number of vehicles, because that the number of couriers as well as their daily tasks is fixed. In view of this problem, we utilize the Push Forward Insertion Detection Method (PFIDM) [8] to allow users to change their appointment time, when a time conflict occurs.

III. Personalized Travel time estimation service

A. Definitions

Definition 1 (Couriers Trajectory): A courier trajectory T is a sequence of time-ordered points, T: $p_1 \rightarrow p_2 \rightarrow \cdots$

$\rightarrow p_n$, where each p_i contains latitude $p_i.\mathrm{lat}$, longitude $p_i.\mathrm{lng}$ and timestamps $p_i.\mathrm{t}$.

Definition 2 (Road Network): A road network is a directed graph G(V, E), where V is a set of vertices representing the intersections and terminal points of the road segments, and E is a set of edges representing road segments.

Definition 3 (Path): A path is represented as a sequence of connected road segments, e.g., P: $e_1 \rightarrow e_2 \rightarrow \cdots \rightarrow e_n$ in Road network.

B. Travel Time Estimation Service Overview

In this section, we discuss the framework of the personalized travel time estimation service in detail. Figure 2 presents the framework of travel time estimation service, which comprises the three major parts.

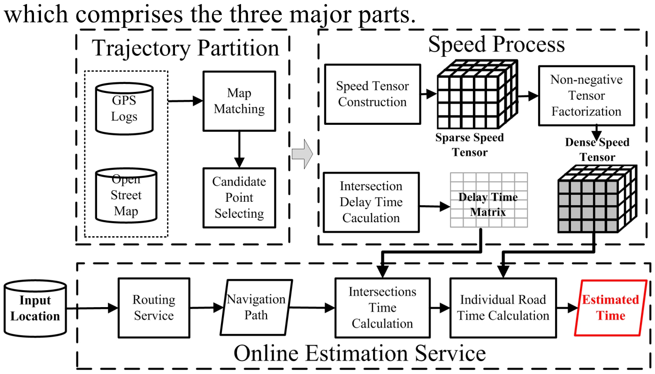

Figure 2. Architecture of Travel Time Estimation

In the trajectory partition part, we partition each courier trajectory into a sequence of connected road segments based on the map-matching algorithm [16]. Then we construct the speed tensor S_s where the three dimensions stand for couriers, road segments and time slots. It is clearly that the tensor is very sparse because a courier can only travel a few road segments in a time slot. In tensor factorization part, to infer the missing values in speed tensor, we follow the idea of tensor factorization that there are some latent features underlying the interactions among users, time slots and road segments in the speed tensor. After filling the missing speed in S_s, we can obtain the dense speed tensor S_d, which stores the travel speed of any courier on any road segment in specific time slot. In the online estimation service, we use routing service to get the navigation path P from the start location to end location. With the path P, we estimate the travel time for path P in the current time slot by summing up the travel times of the road segments belonging to path P based on the travel speed tensor.

C. Trajectory Partition

Given a courier trajectory T $= p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, we need to partition the trajectory into connected road segments. For example, T will be partitioned into edges ($e_1 \rightarrow e_2 \rightarrow \cdots \rightarrow e_n$), each edge e_i contains the travel length $e_i.\mathrm{len}$ and time interval $e_i.\mathrm{t}$ on the road segment. With the partitioned edges, we can calculate the average speed on the road segments in the current time slot for the courier. However, because there are some GPS positioning errors for p_i, we first need to map the point p_i to point c_i on the road segments.

1) Map Matching

To get the candidate point c_i for point p_i, we first get the connected road segments ($e_1 \rightarrow e_2 \rightarrow \cdots \rightarrow e_n$) for the trajectory T based on the map-matching algorithm[17]. For example, as the Figure 3 shows, for the trajectory ($p_{i-1} \rightarrow p_i \rightarrow p_{i+1}$), we can get the connected road segment ($e_i^1 \rightarrow e_i^2$). After getting the connected road segments, we need to map the point p_i to the road segment. However, it's not easy for us to select the right candidate point. For example, for the point p_i and the road segments e_i^1 e_i^2 , the candidate points c_i^1 c_i^2 are both possible. Therefore, we need to select the best candidate point c_i^j for point p_i, as shown in Figure 3.

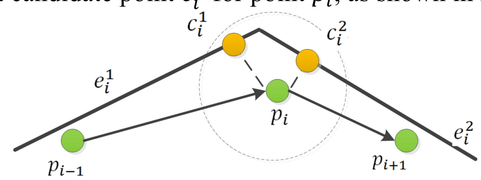

Figure 3. An example of selecting candidate points

2) Candidate Point Selecting

Given the candidate points $(c_i^1, c_i^2, \cdots, c_i^n)$ for the point p_i on the mapped road segments, we will use geometric and topological information of the road network to choose the best candidate. The geometric analysis is mainly based on the distance between c_i^j and p_i. Generally speaking, the smaller the distance is, the better the candidate point maps. Therefore, the positioning error can be described as a normal distribution $N(\mu, \sigma^2)$ of the distance between (c_i^j , p_i). We formally define the geometric probability of c_i^j as:

$$G\left(c_i^j\right) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{(x_i^j-\mu)^2}{2\sigma^2}}$$

where x_i^j is the distance between c_i^j and p_i.

The geometric analysis doesn't consider the neighboring points. For example, as Figure 3 shows, although p_i is close to c_i^2, we should match p_i to c_i^1 if we know it is impossible for the courier from p_{i-1} to c_i^2. To solve this problem, we assume that the speed on the road segment can be also described as a normal distribution[22]. The transaction probability from p_{i-1} to c_i^j can be formalized as :

$$V\left(p_{i-1}, c_i^j\right) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{(s_i^j-\mu)^2}{2\sigma^2}}$$

where s_i^j is the speed from p_{i-1} to c_i^j. In this paper, we assume the average speed for the courier is 4 m/s and the deviation of 2 m/s based on empirical evaluation.

Considering the geometric and topological analysis, we can get the possibility for each candidate:

$$P\left(c_i^j\right) = G\left(c_i^j\right) * V\left(p_{i-1}, c_i^j\right).$$

In this paper, we will choose the candidate point with the biggest possibility.

D. Dealing with missing speeds

1) Intersection Delay Time Matrix

In this paper, we apply the idea of estimating the travel time of individual road segments and then summing up the travel times of the road segments belonging to a path. Therefore, to model the complex factors for crossing two road segments, we construct the 2D matrix M, which represents the intersection delay time from one road segment to the next road segment. As Figure 4 shows,

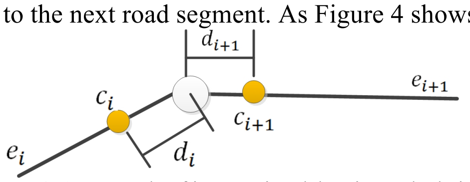

Figure 4. An example of intersection delay time calculation

The intersection delay time from e_1 to e_2 can be calculated as:

$$M_{(e_i \to e_{i+1})} = T_{c_{i+1}} - T_{c_i} - \frac{d_{i+1}}{s_{e_{i+1}}} - \frac{d_i}{s_{e_i}}$$

where, T_{c_i} represents the timestamp at point c_i and s_{e_i}represents the speed on road segment e_i.

2) Speed Tensor Building

After calculating the intersection delay time, in order to estimate the travel time of individual road segments, we need to calculate the travel speed. The partitioned trajectories can be used to construct a 3D speed tensor S_s, which represents the speed on the specific road segment in the specific time slot for the specific courier. We partition a day into three time slots based on the degree of congestion (low, middle, high). Meanwhile, because a courier is responsible for a specific region, we partition the road networks in Beijing into 400 regions and each courier belongs to one region. Clearly, the tensor S_s is very sparse because a courier can only travel a few road segments in a time slot. To deal with the missing speed values in the tensor, we apply non-negative tensor factorization [18] to fill the missing values.

3) Non-Negative Tensor Factorization

Let us denote the three-way speed tensor $S_s = \{s_{ijk}\} \in \mathbb{R}^{I*J*K}$ with only non-negative values $s_{ijk} \geq 0, \forall\, i, j, k$. Figure 5 illustrates the concept of NTF. NTF factorizes a tensor S into three matrices, each of which consists of R factors and only contains non-negative values. We denote the three matrices as T = $\{t_{ir}\} \in \mathbb{R}^{I*R}$, U = $\{u_{jr}\} \in \mathbb{R}^{J*R}$ and V = $\{v_{kr}\} \in \mathbb{R}^{K*R}$. The r-th column vectors of each matrix correspond to the r-th factor of the tensor.

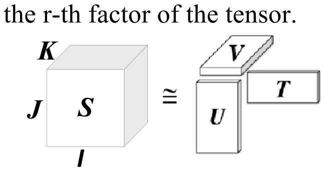

Figure 5. An example of non-negative factorization

With T, U, V, we can reconstructed the tensor S_s as $S_d = \{s_{ijk}\} \in \mathbb{R}^{I*J*K}$. We define s_{ijk} as the sum of the linear products of the three matrices.

$$s_{ijk} = \sum_{r=1}^{R} t_{ir} u_{jr}\, v_{kr}$$

Therefore, to fill the missing value in sparse tensor S_s, we need to calculate T, U, and V. Then the NTF is formulated as follows:

$$\min D(s_s | s_d; \Theta) \;\; subject\ to\ T, U, V \geq 0$$

Where $\Theta \triangleq \{U, T, V\}$. D denotes the divergence between two tensors.

In this paper, we use the generalized Kullback-Leibler (gKL) divergence, which is formulated as:

$$d(p|q) = -plogq + q + plogp - q\ ,$$

and multiplicative update algorithm to solve the NTF optimization problem [19]. Therefore, after process with the NTF method, we can construct the dense speed tensor S_d, which stores the travel speed of any courier on any road segment in specific time slot.

E. *Online Estimation Service*

In this section, we present the online estimation process, which is based on the generated speed tensor S and intersection delay time matrix M.

Algorithm 1. Online Estimation Service

Input: Active courier: u, start location: s, destination: e, current slot: t

Output: The estimated travel time from s to e

Definitions: Couriers' travel speed tensor S, intersection delay time matrix between road segments M.

```
1: T = 0       // Travel Time
2: Roads ⟵ getNavigatePath(s, e)
3: For each r ∈ Roads do
4:      r_n ⟵ next road for r in Roads
5:      T  ⟵ T + r.length / S[u][r.no][t]
6:      T  ⟵ T + M[r.no][r_n.no]
7: End For
8: return T
```

Algorithm 1 illustrates the procedure of the online estimation service. Line 2 gets the navigation path from start location to end location using online routing services [20]. The routing result consists of the key points on the road segments and the road segments ($e_1 \to e_2 \to \cdots \to e_n$) belong to the path. Line 3-6 calculate the travel time for the path. For each road segment r, we first calculate the travel time on this road segment by using the travel length to divide the travel speed in speed tensor S. Second, we add the intersection delay time from r to the next road segment r_n in M. With the travel time for individual road segments, we can get the travel time by summing up the travel times of the road segments belonging to the path.

IV. EXPRESS DELIVERY SCHEDULING SERVICE

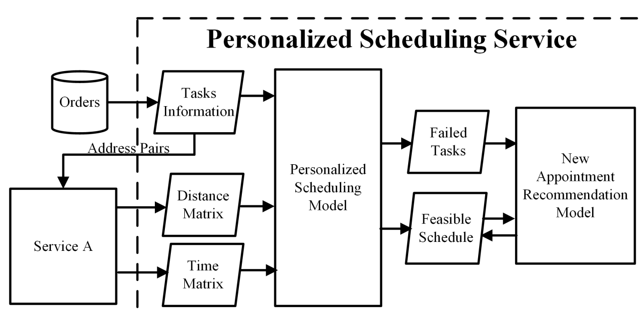

Figure 6. Framework of express delivery scheduling service

Figure 6 shows the framework of the proposed express delivery scheduling service, which is composed of two essential parts. In the first part, prior to the main scheduling process, it is necessary to get the courier's one-day task information, and request the distance and the personalized travel time between each pair of tasks from Service A, i.e. personalized travel time estimation service proposed in section III. Using these data, the personalized scheduling model can compute a feasible delivery schedule and get a set of conflicted tasks. Then, in the second part, a new appointment recommendation model is put forward to solve the conflicted tasks. The rest of this section gives the detailed descriptions of these two models mentioned above.

A. Personalized Scheduling Model

Nowadays, in the "last mile" delivery, it's very common for a courier to be assigned with delivery tasks whose addresses are located in a specific region. Usually he uses his experience (familiarity with the addresses, roads in this region) to schedule and deliver these tasks. However, at present, more and more customers want to receive their goods in their expected time windows rather than whenever the courier arrives. Thus, the courier cannot schedule these tasks with different time windows efficiently, only using his experience. So we propose this personalized scheduling service to help couriers to make a feasible schedule when they are facing such a problem.

1) Problem Formultaion

The express delivery scheduling problem mentioned above, in its simplest form, can be formulated as a VRPTW. However, considering the real-life factors, such as the number of couriers who deliver the goods in a region is limited to one, the V.I.P. levels of the customers are different, etc., there are some specific changes from the traditional VRPTW. We formulate our problem as following.

Let $C = \{c_0, c_1, c_2, \dots, c_n\}$ represent a set, including one depot(c_0) and n customers $(c_1, c_2, \dots c_n)$. Let $d(c_i, c_j)$ and $t(c_i, c_j)$ separately represent the distance and the travel time on the path between c_i and c_j, arc(c_i, c_j), $c_i, c_j \in C$. Besides the address, each customer c_i involves a V.I.P. level lv_i and a service time s_i, which represents the time required for c_i to sign the delivered goods. Moreover, there is a time window $[b_i, e_i]$ during which the service of c_i must be started (the vehicle can arrive before the time window opening, but it is forced to wait until the opening moment) , meanwhile, the depot(c_0) only has a begin time b_0. In addition, there is only one vehicle and it should set out from the only depot at b_0 with a constrained capacity W. The solution is a scheduled route, $R = (c_0', c_1', c_2', \dots, c_n')$ where $c_i' \in C$, $c_0' = c_0$ and $c_i' \neq c_j'\ if\ i \neq j$, in which a customer is served exactly once. And the capacity of the vehicle shouldn't be exceeded. Finally, the goal is to find a feasible route which is minimized in two criteria： the number of the conflicted customers whose services cannot start in their time window and the total travel time.

2) Applying An Improved Ant Colony System (ACS)

ACS is a meta-heuristic algorithm that inspired from the behavior of ants, where ants cooperate to search for the best path towards foods by depositing pheromone along the way. In computational implements, artificial ants constructing the route simulates real ants searching the path, the objective function corresponds to the quality of the path to foods, and the numerical pheromone corresponds to the real pheromone trails. The base process of the algorithm is shown as follow, where the elitist ants are a fixed number of top ranked ants.

Algorithm 2. Improved Ant Colony System

Input: $C = \{c_0, c_1, c_2, \dots, c_n\}$, $d(c_i, c_j)$, $t(c_i, c_j)$, l_i, s_i, $[b_i, e_i]$ // formulated in *1)*.

Output: A feasible route schedule R' and a set of conflicted customers $Conf$.

Definitions: Number of iterations I, number of ants m, ant k 's constructed route R^k which consists of all customers, the best route solution R, pheromone between c_i and c_j τ_{ij}

```
For i = 1 to I do                      // Iterations
    For k = 1 to m do                  // Ants
        R^k ← ant k constructs its route   // Detailed in a)
        Evaluate R^k, update elitist ants  // Detailed in b)
    End For
    R ← update the best route solution
    τ_ij ← update the global pheromone // Detailed in c)
End For
Conf ← extract the set of conflicted customers
R' ← R − Conf
return {R', Conf}
```

a) Route Construction

In our improved ACS, m ants construct their routes in parallel. Each ant starts at the only depot and adds customers into its route one by one, until all the customers have been visited.

In the process, when an ant k finishes the service of the customer c_i, it has a heuristic visibility to customer c_j which is represented by η_{ij} stated as follows:

$$\eta_{ij} = (\frac{1}{d_{ij}})^{\gamma}(\frac{1}{t_{ij}})^{\delta}(\frac{1}{pe_j})^{\zeta}(lv_j)^{\theta}(\frac{1}{ww_j})^{\omega}$$

Here, d_{ij} and t_{ij} are the short terms of $d(c_i, c_j)$ and $t(c_i, c_j)$ defined in 1), pe_j is the penalty of the arrive time to customer c_j, lv_j is the V.I.P. level of c_j and ww_j is the width of c_j's time window, $\gamma + \delta + \zeta + \theta + \omega = 1$. In addition, the penalty function of pe_j is shown in Figure 7.

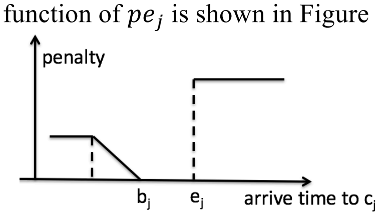

Figure 7. An example of penalty function for time window

Besides the visibility η_{ij} , τ_{ij} which represents the pheromone density between c_i and c_j, and a probability P_0 are used to choose next feasible customer in C_i^k (the set of ant k's unvisited customers at c_i). The specific transition rule is following: with probability of exploitation P_0, the customer $c_j \in C_i^k$ with the highest $[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}$ is chosen, while with probability of exploration $(1 - P_0)$, the customer $c_j \in C_i^k$ is chosen with a probability p_{ij} which is proportional to $[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}$. And the probability p_{ij} can be stated as follows:

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h \in C_i^k} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta} & if\ c_j \in C_i^k \\ 0 & otherwise \end{cases}$$

The parameters α and β represent the biases for pheromone and visibility respectively, determining the relative influence of them in the transition rule.

b) Objective Function

In order to evaluate ant k's route solution, we compute its score sc_k by the objective function below:

$$sc_k = (tt_k)^\lambda (\sum_{i \in R^k} (lv_i^k \cdot pe_i^k))^\mu$$

Here, tt_k is the total travel time of ant k and $lv_i^k \cdot pe_i^k$ means how serious the penalty of customer c_i is. The parameters λ and μ represent the biases for total travel time and total penalty. The smaller score an ant has, the better solution it gets. So the elitist ants of an iteration and also the global best solution could be selected easily.

c) Pheromone Update

The pheromone is like a 'prefered route' memory, and future ants can use this information to generate a new solution in the neighborhood of the prefered route. In our improved ACS, we only update the pheromone globally, that is to say, the pheromone τ_{ij} will only be updated after a complete iteration. Let the number of elitist ants be m_e and each of them has its real rank σ from 1 to m_e. The pheromone on each arc(c_i, c_j) used by one of the elitist ants k, is increased by $\Delta_{ij}^k = \frac{(m_e+1-\sigma)\cdot Q}{sc_k}$, where Q is a constrant parameter, meanwhile, Δ^k on the other arcs equals zero. In addition, if arc(c_i, c_j) belongs to the so far best solution with objective value sc^*, its increased pheromone Δ_{ij}^* equals to $\frac{Q'}{sc^*}$ and zero otherwise, where Q' equals to several times of Q. More over, part of the pheromone will evaporate after an iteration, and the retainability is named as ρ, where $0 < \rho < 1$. So the update rule of the pheromone is stated as follows:

$$\tau_{ij}^{new} = \rho \cdot \tau_{ij}^{old} + \sum_{k=1}^{m_e} \Delta_{ij}^k + \Delta_{ij}^*$$

B. New Appointment Recommendation Model

For both couriers and customers, the conflicted tasks must be eliminated. Our express delivery scheduling service provides the conflicted customers with the service to make new appointments. When there is a conflicted customer, we first examine the insertions between every two successive customers to find the feasible ones for him. Then we compute new time windows for such feasible insertions. Finally, we recommend the conflicted customer some new appointments in a ranked order. The definitions formulated in part A are still used in the following.

1) Insertion Examination

Given a conflicted customer c_u', $c_u' \in Conf$, and the feasible route R', $|R'| = n'$, we adapt Push Forward Insertion Detection Method (PFIDM), to detect whether the insertion of c_u' between c_{i-1}' and c_i', $1 \le i \le n'$, is feasible. Obviously, the insertion at the tail of R' is always feasible.

PFIDM proposes some concepts: according to time window rules, the courier can arrive at a customer c_i at time at_i, but he can only start the service at time st_i within time window $[b_i, e_i]$, so there may be a waiting time wt_i. After the service time s_i, if the courier travels directly from c_i to c_j and arrives too early at c_j, he must wait, that is, $st_j = \max\{b_j, at_j\}$ where $at_j = st_i + s_i + t_{ij}$.

When inserting a customer c_u' between c_{p-1}' and c_p', $1 \le p \le n'$, st_p^{new} denotes the new time when service begins at c_p', given the insertion of c_u'. PFIDM defines a *push forward* at c_p' because of this insertion:

$$PF_p = st_p^{new} - st_p \ge 0$$

Furthermore,

$$PF_{r+1} = \max\{0, PF_r - wt_{r+1}\}, p \le r \le n' - 1.$$

Then it proves the necessary and sufficient conditions for time feasibility when inserting a customer c_u' between c_{p-1}' and c_p', $1 \le p \le n'$, on the feasible route R' is:

$$st_r + PF_r \le e_r, p \le r \le n'$$

According to the method above, we can find out all feasible insertion points for the conflicted customer c_u' on scheduled route R' and we use $I_u = \{(c_{i-1}', c_i') | between\ which\ c_u'\ can\ be\ inserted\}$, to represent the set of such points.

2) Computing Feasible Time Windows

After finding all the feasible insertion points for the given conflicted customer, we now propose the methods to compute new feasible time windows. When c_u' is inserted between c_{p-1}' and c_p', $1 \le p \le n'$, there will be a new time window of c_u', $[b_u^{new}, e_u^{new}]$, which represents c_u''s earliest and the latest feasible service start time.

On the basis of the method proposed in 1), it is easy to figure out $b_u^{new} = st_{p-1} + s_{p-1} + t_{p-1,u}$. Because the service start time at c_u' can't be earlier than at_u, that is to say, the earliest feasible service start time at c_u', the b_u^{new}, equals to at_u, where $at_u = st_{p-1} + s_{p-1} + t_{p-1,u}$.

Then we formulate the latest feasible service start time at customer c_i' as st_r^{latest}, which means any later start time at c_i' will cause some scheduled customer after c_i' infeasible. And we define an *allowable delay* for customer c_r', which represents the maximum delay that is allowed between the latest feasible start time and the arrive time at c_r', as follows:

$$AD_r = st_r^{latest} - at_r \ge 0$$

And the update rule of st_r^{latest} is defined in following:

$$st_r^{latest} = \begin{cases} e_r & , \quad r = n' \\ \min\{st_r + AD_{r+1}, e_r\}, & p \le r \le n' - 1 \end{cases}$$

That is to say, the tail customer's latest feasible start time is the end of his time window. And the expression $\min\{st_r + AD_{r+1}, e_r\}$ can be understood as "customer c_{r+1}' tells c_r' that you can delay your start time at most AD_{r+1}, and then c_r' judges whether he can start at $st_r + AD_{r+1}$ or the ending time of his time window e_r". Thus, we can get the latest feasible start time st_p^{latest} at customer c_p'. Furthermore, we can figure out that $e_u^{new} = st_p^{latest} - t_{u,p} - s_u$.

So far, each feasible insertion point belonging to I_u has a new feasible time window.

3) Recommending new appointments

Considering the cost of each feasible insertion in I_u is different, the new recommended appointments are ranked by the following cost function:

$$cost_i = (t_{i-1,u} + t_{u,i})^{\iota} \cdot (d_{i-1,u} + d_{u,i})^{\kappa} \cdot (ww_u^{new})^{\varepsilon},$$
$$(c_{i-1}', c_i') \in I_u$$

Here, $\iota + \kappa + \varepsilon = 1$, and the smaller cost means the higher rank.

Finally, we recommend the ranked feasible appointments to the conflicted customer c_u'.

V. Experiments

Travel time estimation service and express delivery scheduling service compose the personalized service for scheduling express delivery.

We evaluate the effectiveness and efficiency of the two major services of travel time estimation and express delivery scheduling as follows.

A. Travle Time Estiamtion Results

To evaluate the effectiveness of the proposed method named as CTE (Courier Travel Estimation), we compare it with the following two baseline methods, which are used widely in real world.

1) Navigation-Service-based (NS) method. The path travel time is estimated by a popular navigation service named as Baidu Maps [21], which is widely used in China.
2) Average-Speed-based (AS) method. The travel time of each road segment is computed by the length of a road segment and its average speed.

We use a GPS trajectory datasets collected by 210 couriers in Beijing lasting for ten days. The number of GPS points reaches 1,195,431, and the total length of the trajectories is over 54,243 km. The average sampling rate is 30 seconds per point.

We randomly pick out 20 couriers, and select 20 paths in the trajectory for each courier. We then use these paths as queries and the time interval in the trajectory as the ground truth. Once the trajectories are selected as the ground truth, we will remove them from the training data. In total, we generate 854 queries, whose path duration range from 2min to 12min. In the experiments, we only consider the trajectories between 9am to 6pm, as this is the work time for the courier.

We study the mean absolute error (MAE) and mean relative error (MRE) as follows:

$$\text{MAE} = \frac{\sum_i |y_i - \hat{y}_i|}{n},$$
$$\text{MRE} = \frac{\sum_i |y_i - \hat{y}_i|}{\sum_i \hat{y}_i}.$$

Where y_i is the estimated result and $\hat{y}_i$ is the ground truth.

Table I presents the overall performance of the proposed method and the baseline method with the path duration is 10min. Besides the MAE and MRE, we also present the average error of the travel time per km (MAE/L).

TABLE I. Comparasion of different methos

	MAE (min)	*MRE*	*MAE/L (min/km)*
NS	4.18	0.664	2.797
AS	2.95	0.455	1.974
CTE	**2.00**	**0.308**	**1.338**

Clearly, CTE outperforms all the baselines in terms of the three metrics. From the results, we can draw the following conclusions. First, using AS is much better than using NS because the NS can only provide the travel time by car or on foot, which is not suitable for the courier. Therefore, comparing the AS with the NS, we can demonstrate the necessity of the travel time estimation. However, simply using the average speed for calculating the travel time is not an optimal solution. As we have mentioned, different couriers have different travel speeds on a specific road segment. Therefore, comparing with the AS can reveal the advantages of using the couriers' trajectory data.

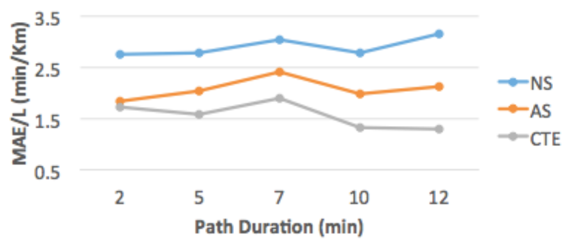

Figure 8 MAE/L for different path duration

Figure 8 presents the MAE/L error changing over the path duration. As depicted in Figure 8, after 7min, with the path duration increasing, the MAE/L using CTE decreases because the shorter path is easily influenced by some uncertain factors. What's more, with the path duration increasing, the difference between CTE and AS increases.

B. Express Delivery Scheduling

The experiment of the express delivery scheduling service contains two steps. The first step of the experiment is conducted by analyzing the influence of a single variable as well as the interaction among all the variables, and dividing them into the following independent groups listed from global to local: a) iteration count I, b) the number of ants m and elitist ants m_e, c) pheromone evaporation rate ρ and accumulation factor Q, d) probability of exploitation P_0, e)the biases of pheromone α, f) weight of parameters in the heuristic visibility function. These groups are independent of each other, while interdependent within each group.

TABLE II. Optimized Variables

Variable	*Value*	*Variable*	*Value*
$\boldsymbol{I}$	90	$\boldsymbol{Q}$	300
$\boldsymbol{m_e/m}$	11/95	$\boldsymbol{P_0}$	0.67
$\boldsymbol{\rho}$	0.9	$\boldsymbol{\alpha}$	0.16

The optimized values shown in Table II are found by enumerating with reasonable step length and utilizing the control variable method between different groups.

The second step of experiment is designed to compare our model with the random algorithms and the greedy algorithms that simulate real couriers. Two factors are considered in the evaluation of the algorithms' performance: Conflict Score, which equals the sum of $\log_{10} VLC$ (this factor should be readjusted based on specific application), where VLC is the VIP level of a conflicted customer; Finish Time, which is the time when all non-conflicted delivery tasks are completed.

TABLE III. COMPARISON EXPERIMENT SETUP

Name	*Value*	*Name*	*Value*
Task #	20	**Service Time (s_i)**	3-6 (min)
VIP # (L1:L2:L3:L4)	7:7:5:1	**Time Window ($[b_i, e_i]$)**	$8{:}30 \le b_i \le e_i \le 12{:}00$

TABLE IV. COMPARISON RESULTS

Algorithm	*Conflict Score*	*Finish Time*
Algorithm in this paper	1.26	11:41
Greedy(Ordered by distance)	9.18	11:44
Greedy(Ordered by time window end time)	5.13	11:46
Random(Best in 10000 times)	6.35	11:37

The basic setup of the experiment is listed in Table III, and we repeated the experiment with five different data sets. And the average results are shown in Table IV. From the results, it is proved that the model in this paper can reach a relatively optimized result within reasonable runtime.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a personalized service for scheduling express delivery using courier trajectories in order to minimize the total time of a courier's one-day delivery. In this service, we design two basic services: a) personalized travel time estimation service for any path in express delivery using courier trajectories, b) an express delivery scheduling service considering multiple factors, including customers' appointment, one-day delivery costs, etc., which is based on the accurate travel time estimation in the first service. The experiments verify the effectiveness and efficiency of the two major services of travel time estimation and express delivery scheduling in our approach.

In the future work, we will continue improving the performance of our scheduling algorithm by applying the service in real situations and collecting feedbacks from both couriers and customers.

ACKNOWLEDGMENT

This effort is sponsored by the High-Tech Research and Development Program of China under Grant No.2015AA01A202, the National Natural Science Foundation of China under Grant No.61421091, No.U1201252.

REFERENCES

[1] Lee, Hau L., and Seungjin Whang. "Winning the last mile of e-commerce." *MIT Sloan Management Review* 42.4 (2001): 54.

[2] Mori, Usue, et al. "A review of travel time estimation and forecasting for Advanced Traveller Information Systems." *Transportmetrica A: Transport Science* 11.2 (2015): 119-157.

[3] Wang, Yilun, Yu Zheng, and Yexiang Xue. "Travel time estimation of a path using sparse trajectories." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.

[4] De Fabritiis, Corrado, Roberto Ragona, and Gaetano Valenti. "Traffic estimation and prediction based on real time floating car data." *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008.

[5] Jenelius, Erik, and Haris N. Koutsopoulos. "Travel time estimation for urban road networks using low frequency probe vehicle data." *Transportation Research Part B: Methodological* 53 (2013): 64-81.

[6] Wang, Ke. "Research on the Selection of the Third Party Logistics' Distribution Center." *2014 International Conference on Mechatronics, Control and Electronic Engineering (MCE-14)*. Atlantis Press, 2014.

[7] Wu, Chengyong, et al. "A Study on Choosing Sites for Logistics Distributions Center Based on GIS Service Area Analysis." *Proceedings of the 2013 International Conference on Information System and Engineering Management*. IEEE Computer Society, 2013.

[8] Solomon, Marius M. "Algorithms for the vehicle routing and scheduling problems with time window constraints." *Operations research* 35.2 (1987): 254-265.

[9] Hong, Sung-Chul, and Yang-Byung Park. "A heuristic for bi-objective vehicle routing with time window constraints." *International Journal of Production Economics* 62.3 (1999): 249-258.

[10] Banos, Raul, et al. "A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows." *Expert Systems with Applications* 40.5 (2013): 1696-1707.

[11] Moccia, Luigi, Jean-François Cordeau, and Gilbert Laporte. "An incremental tabu search heuristic for the generalized vehicle routing problem with time windows." *Journal of the Operational Research Society* 63.2 (2012): 232-244.

[12] Vidal, Thibaut, et al. "A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows." *Computers & Operations Research* 40.1 (2013): 475-489.

[13] Balseiro, S. R., Irene Loiseau, and Juan Ramonet. "An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows." *Computers & Operations Research* 38.6 (2011): 954-966.

[14] Qi, Chengming, and Yunchuan Sun. "An improved ant colony algorithm for VRPTW." *Computer Science and Software Engineering, 2008 International Conference on*. Vol. 1. IEEE, 2008.

[15] Selvi, V., and Dr R. Umarani. "Comparative analysis of ant colony and particle swarm optimization techniques." *International Journal of Computer Applications (0975–8887)* 5.4 (2010).

[16] Lou, Yin, et al. "Map-matching for low-sampling-rate GPS trajectories." *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009.

[17] Yuan, Jing, et al. "An interactive-voting based map matching algorithm." *Proceedings of the 2010 Eleventh International Conference on Mobile Data Management*. IEEE Computer Society, 2010.

[18] Takeuchi, Ken, et al. "Non-negative multiple tensor factorization." *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 2013.

[19] Lin, Chih-Jen. "On the convergence of multiplicative update algorithms for nonnegative matrix factorization." *Neural Networks, IEEE Transactions on* 18.6 (2007): 1589-1596.

[20] https://github.com/graphhopper/graphhopper/

[21] http://map.baidu.com/

[22] Dey, Partha Pratim, Satish Chandra, and S. Gangopadhaya. "Speed distribution curves under mixed traffic conditions." *Journal of transportation engineering* 132.6 (2006): 475-481.