CrowdExpress: a probabilistic framework for on-time crowdsourced package deliveries

Chao Chen, Sen Yang, Yasha Wang, Bin Guo, Daqing Zhang

▶ **To cite this version:**

Chao Chen, Sen Yang, Yasha Wang, Bin Guo, Daqing Zhang. CrowdExpress: a probabilistic framework for on-time crowdsourced package deliveries. IEEE Transactions on Big Data, 2022, 8 (3), pp.827-842. 10.1109/tbdata.2020.2991152 . hal-03002740

HAL Id: hal-03002740
https://hal.science/hal-03002740v1

Submitted on 12 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CrowdExpress: A Probabilistic Framework for On-Time Crowdsourced Package Deliveries

Chao Chen, Sen Yang, Yasha Wang, Bin Guo and Daqing Zhang

***Abstract*—Most of current urban logistic systems fail to strike a nice trade-off between speed and cost. An express logistic service often implies a high delivery cost. Crowdsourced logistics is a promising solution to alleviating such contradiction. In this paper, we propose a new form of crowdsourced logistics that organizes passengers and packages in a shared room, i.e., using taxis that are already transporting passengers as package hitchhikers to achieve on-time deliveries. It is well-recognized that taxi drivers are good at delivering passengers to their destinations efficiently. As a result, the proposed new urban logistics system has potentials to lower the cost and accelerate package deliveries simultaneously. Specifically, we propose a probabilistic framework containing two phases called CrowdExpress for the on-time package express service. In the first phase, we mine the historical taxi GPS trajectory data *offline* to build the package transport network. In the second phase, we develop an *online* taxi scheduling algorithm to adaptively discover the path with the maximum arriving-on-time probability "on-the-fly" upon real-time passenger-sending requests, and direct the package routing accordingly. Finally, we evaluate the system using the real-world taxi data generated by over 19,000 taxis in a month in the city of New York, US. Results show that around 9,500 packages can be successfully delivered daily on time with the success rate over 94%.**

***Keywords*—package delivery; shared mobility; hitchhiking rides; route planning; taxi scheduling; trajectory data mining**

I. Introduction

Recent years have witnessed an exponentially growth of online orders, as a result of the pervasive use of tablet and mobile devices, which urges the sustainable development of urban logistics industry [12], [24]. People present an increasing number of diverse requirements on the urban delivery service, e.g., the express service as a requirement of the fast speed, the contactless one as a requirement of the epidemic prevention during the current coronavirus crisis[1]. To online shoppers, *speed and cost* are still the two major concerns, which are usually conflicted with each other in nature. Urban crowdsourced logistics (also termed as crowdshipping), with the idea of completing package deliveries from *little or no* effort from the crowd, is recognized as a promising solution to alleviating such contradiction. Hence, in this paper, we propose a similar idea, which leverages the collaborative efforts from the crowd of taxis and sends packages together with passengers in a shared space and transport network. In line with the previous research with a particular focus on taxis, we propose having packages take *hitchhiking rides* collaboratively with existing taxis that are transporting passengers on the street, i.e., the existing mobility of taxi drivers [4], [23]. In more detail, we target to design a specific crowdsourced logistics called CrowdExpress that can lower the cost but still ensure the *on-time arrivals* of packages [6], [28]. We argue that, compared to the previous work which aims to discover the optimal package-sending route in terms of the least delivery time cost, it is more common and reasonable in real cases that online shoppers are actually more concerned about whether their package can arrive at their homes by the deadline or not, and show little interest at the specific arriving time.

Considering the quality of service to passengers should be the top-priority for taxi drivers, the proposed CrowdExpress only requires small additional efforts and time from the taxi drivers involved, without degrading the service quality much and any interrupt to passengers. As discussed in our previous study, we can formulate the proposed taxi-based package delivery as a route planning problem, but with a quite different objective, i.e., delivering the packages to their destinations on time (by the deadline given by users). To make the idea of organizing the passenger flow and package flow seamlessly via the taxi transport network feasible, we need to address the following two major research challenges:

- Package flow and passenger flow are *incompatible* in both time and space. In more detail, 1) compared to the package flow, the passenger flow presents salient peak-hour patterns; 2) due to the financial considerations, most passengers choose to take taxis only when the destination is close, e.g., within 4 km [8]. While for packages, the destination is generally far away from the origin (e.g., longer than 5 km) [4]. Therefore, we argue that routing algorithms based on the framework of direct *query-matching* merely may not work [14], [21], since a single hitchhiking ride may not be able to deliver a package to its destination; instead, a collaborative relay of taxis is needed.
- Requests for demands of packages and passengers come in stream with high *uncertainties*. Although regular spatial and temporal patterns about the passenger flow have been unveiled from taxi GPS trajectory data in the coarse granularity, it is still challenging to predict the passenger demands accurately in a quite fine granularity (e.g., the passenger demands in the next 5 minutes for a given

Chao Chen and Sen Yang are with the Key Laboratory of Dependable Service Computing in Cyber Physical Society (Chongqing University), Ministry of Education and also with the College of Computer Science, Chongqing University, Chongqing 400044, China. E-mail: cschaochen@cqu.edu.cn.

Yasha Wang and Daqing Zhang are with the Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China. E-mail: {dqzhang,wangys}@sei.pku.edu.cn.

Bin Guo is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China. E-mail: guobin.keio@gmail.com.

[1]https://nyti.ms/33PqNWx

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

road segment). Meanwhile, same situation also happens to package demands [32]. In a word, uncertainties exist in both requests, making the estimation and comparison of time cost of package routing paths challenging.

With the above-mentioned research objective and challenges, the main contributions of the paper are:

1) We propose a novel modification of crowdsourced logistics, i.e., passenger and package mixed transport mode which leverages the *unintentional cooperations* among a crowd of passenger-occupied taxis to meet city-wide on-time package deliveries, in order to lower the transport cost and enhance the transport efficiency simultaneously. Moreover, the package handover (uploading and offloading) takes place by taxi drivers before picking up (after dropping off) passengers, thus the service quality to passengers can be guaranteed.
2) We formulate the package routing problem as the arriving-on-time problem [26] to tackle with uncertainties of passenger and package requests. We propose a probabilistic two-phase framework named CrowdExpress to solve it. In the first phase, we build the package transport network by mining the historical taxi GPS trajectory data *offline*. In the second phase, for each real-time generated package delivery request, we propose an *online adaptive taxi scheduling algorithm* based on the probabilistic model called **maxProb** to iteratively determine the next stop of the package "on-the-fly". The algorithm monitors real-time taxi ordering requests, *recursively* computes the maximum arriving-on-time probability if assigning the delivery task to the currently available taxi, and compares it to the one if waiting for future taxi hitchhiking rides, based on the real-time package location and the remaining time budget.
3) We conduct extensive evaluations using road network data and taxi GPS trajectory data generated over 19,000 taxis in a month in the city of New York (NYC) to verify the efficiency and effectiveness of CrowdExpress. Results demonstrate that it responds within 25 milliseconds. What is more, it can throughput around 9,500 packages daily with the success rate over 94%, which is consistently better than the baseline approaches.

The rest of the paper is organized as follows. In Section II, we review the related work and show how this paper differs from the prior research. In Section III, we introduce some basic concepts, the assumptions we have made, the formal problem formulation, and overview the system. We present the technical details about our two-phase approach in Section IV and Section V respectively. We evaluate the performance of the proposed framework in Section VI. Finally, we conclude the paper and chart the future directions in Section VII.

II. Related Work

Crowdsourcing has been used for many different applications, from problem solving and various urban sensing tasks to the last-mile delivery [10]. Some papers also recognized the crowdsourced logistics as a specific kind of spatial crowdsourcing tasks [15], [23]. Early efforts on crowdsourced logistics including two representative papers which leverage the spatial and time overlaps between crowdsourcing workers. Specifically, Sadilek et al. [30] recruited a group of Twitter users, asking one person to pass the assigned package to another Twitter user that happened to be nearby (within a certain distance). However, this work had two main limitations: 1) It is hard to trace and coordinate the users since people rarely share their location information continuously via geo-twitter [29]. Therefore, the choice of suitable deliverers is limited, probably resulting in longer and uncontrollable package delivery delay. 2) It may be not practical to ask a participant to make a dedicated trip to pass the package to another suitable user, as it may interrupt his/her on-going activities (e.g. having conversation/dinner with friends) that are hard to be inferred from the user's geo-twitters data. Similarly, the work in [25] which employed mobile users based on the overlaps of space and time inferred from cell towers had similar limitations: 1) The cell towers may be sparse in certain areas and thus it is difficult for mobile users to relay packages in that areas. 2) Only with data when people make calls, as a result, the number of mobile users that can be recruited is limited.

Towards a more realistic and reliable form of crowdsourced logistics, there are some papers that intended to leverage the abundant existing passenger-delivery trips to hitchhike packages appeared in these two years [3], [4], [14], [17], [20], [21], [22]. For instance, Chen et al. [7] discussed unique features, key research challenges and potential solutions in their vision paper. There are a bunch of crowdsourced logistics systems based on the shared mobility generated when sending passengers, addressing issues from evaluating its public acceptance and potentially substantial impacts to planning optimal delivery routes [4], [9], [12], [27]. Although the passenger flow and package flow are combined to be transported mixed in these proof-of-concept systems, the authors fail to consider their distinct patterns in both time and space. In more detail, they formulate the problem as the *share-a-ride problem* and insert the package requests into the passenger-delivery trips, which may not able to deliver packages successfully in real cases as we argued previously. To make the matters worse, in their solutions, during the passenger-sending course, taxis have to make several dedicated stops and detours, which degrades the service quality to passengers. At the current stage, the research on package routing in crowdsourced logistics almost completely ignored the multi-criteria design of the package relay network [4], [9], [11], [20]. How to design the package relay network (e.g., the optimal number and location of package interchange stations) is vital and challenging, which should be a separated research issue [2], [34]. In this paper, we exploit existing taxi services to deliver packages [11] and simply cluster frequent pickup and drop-off points to get package interchange stations without any optimization technicals. Packages can be temporary stored at interchange stations in-between rides, and thus no time overlap and pairwise contact between participants is needed.

As discussed, our method requires less effort from the participants and can transport packages over a longer distance.

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

TABLE I. TIME SLOT SLICING.

	Work Days	Rest Days
Night-time hours	00:00-06:59 19:00-23:59	00:00-07:59 19:00-23:59
Day-time hours	09:00-16:59	08:00-18:59
Rush hours	07:00-08:59 17:00-18:59	NG

In addition, we try to minimize the impact on the quality and experience of passenger service, which is similar to our previous work [9]. Despite the high similarity, CrowdExpress is different from CrowdDeliver essentially in *the objective, the proposed solution as well as the evaluation environment*. To be more specific, CrowdExpress aims to send each package to the destination on time (by the deadline required by the user), rather than as quickly as possible. We argue that the objective is more reasonable and matches the real-life application scenarios more frequently. In CrowdExpress, we formulate the problem as the one of finding arriving-on-time paths, and propose a probabilistic framework (i.e., **maxProb**) to address it. Finally, we evaluate the system performance using the real-world taxi trajectory data collected from New York City (NYC). Due to the openness, we choose to use the taxi trajectory data from NYC in our experiments, expecting to lower the data barrier and attract more researchers with different backgrounds into this promising interdisciplinary filed.

III. PRELIMINARY, PROBLEM STATEMENT AND SYSTEM OVERVIEW

In this section, we provide definitions of some basic concepts, elicit assumptions we have made, and give a formal problem statement. Finally, we give an overview of the proposed CrowdExpress system.

A. Preliminary

1) Basic Concepts: We define the basic concepts used in this work as follows:

Definition 1 (Time Slot Slicing)**.** *We divide a whole day into different time slots (periods) according to the day type, since the traffic conditions are changing in different time slots, resulting in large variance in travel time. A work day is divided into three time slots and a rest day is divided into two time slots, as detailed in Table I.*

Definition 2 (Taxi Trajectory)**.** *A taxi trajectory tr is a sequence of GPS points corresponding to a single passenger-delivery trip. Here, the taxi trajectory is represented by a pair of Origin-Destination (OD), where the origin is the road segment that the trip starts and the destination is the one that the trip ends. The travel time is exactly the time difference between the ending and starting times.*

Definition 3 (Package Transport Network)**.** *A package transport network is a graph $G(S, E)$, consisting of a node set S and an edge set E. Each element s in S is an interchange station which is responsible for package collections and storage. Each element $e(i, j)$ in E is a non-stop directional transport route from node s_i to node s_j, implying that there is an abundant passenger flow for hitchhiking packages. It should be noted that the edge in the package transport network has different meaning from the edge defined over the road network. There can be multiple driving paths over the road network that connect two interchange stations, associating with different travel time at different time slots.*

Definition 4 (Real-time Taxi Ordering Request)**.** *A taxi ordering request (tor) is defined as a triplet $\langle o_t, d_t, r_t \rangle$, where o_t and d_t refer to the passenger's origin and intended destination, respectively. r_t refers to the time that the passenger submits the request. The time when the passenger is picked up is usually after r_t.*

Definition 5 (Package Delivery Request)**.** *A package delivery request (pr) is defined as a triplet $\langle o_p, d_p, t_p \rangle$, where o_p and d_p refer to the origin and destination of the package delivery respectively; t_p refers to the time when the user submits the request (i.e. the birth time). Note that here $o_p \in S$, $d_p \in S$, indicating that packages should originate and end at interchange stations.*

Definition 6 (Travel Time Probability Function)**.** *Each edge in the package transport network (G) is associated with an independent random travel time (cost) t_{ij} whose probability density function is denoted by $p_{ij}(t)$. $p_{ij}(t)$ varies at different time slots since traffic conditions vary.*

For instance, the probability that a package spends a time in the interval $[0, t_0]$ from node s_i to node s_j directly can be computed by the definition, as shown in Eq. 1.

$$P_{ij}(t \leq t_0) = \int_0^{t_0} p_{ij}(t)dt \tag{1}$$

The travel time probability function in each time slot can be obtained separately, according to Eq. 1. What is more, as can be observed, the travel time probability is a *monotonic increasing function* of the time t.

Definition 7 (Travel Time Discretization)**.** *To simplify the calculation of travel time probability along an edge, we consider the travel time in a discrete manner. More precisely, we use a piecewise constant function with equal step width τ to discretize different travel times*[2].

In the discrete case, the integral shown in Eq. 1 can be replaced using the formula shown in Eq. 2.

$$P_{ij}(t \leq t_0) = \frac{\#traveltimes < \alpha\tau}{\#traveltimes} \tag{2}$$

$$t_0 = (\alpha - 1)\tau + \delta \tag{3}$$

where $\#traveltimes$ refers to the number of all the possible travel times from node i to node j which are recorded in the taxi trajectory in history, while $\#traveltimes < \alpha\tau$ refers to the number of travel times less than $\alpha\tau$ after time discretization as defined; $\alpha \in N^+$, and $0 \leq \delta < \tau$. As can be seen in Eq. 3,

[2]Here, we set $\tau = 5\ min$ throughout the whole paper.

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

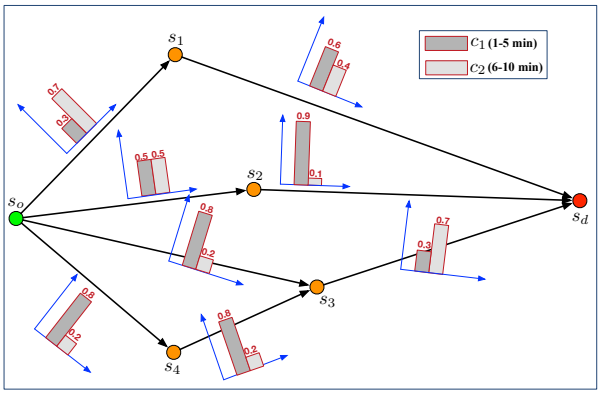


Fig. 1. An illustrative example of package delivery paths from s_o to s_d, as well as the distribution of discrete travel times on each edge.

α and δ are the quotient and reminder of $\frac{t_0}{\tau}$, respectively. For the edge from s_o to s_1 shown in Fig. 1, $P_{o1}(t \leq 5) = 0.3; P_{o1}(5 < t \leq 10) = 0.7$.

Definition 8 (Arriving-on-Time Probability)**.** *The arriving on time probability of a package-delivering path within a given time duration (i.e., deadline, t_0) is defined as the ratio of the number of travel times less than t_0 to the number of all possible travel times (suppose that the package is shipped from s_i to s_j via s_k), as follows.*

$$P_{ij}(t \leq t_0|Path) = \int_{t_0-t_1}^{t_0} dt_2 \int_0^{t_1} p_{ik}(t_1)p_{kj}(t_2 - t_1)dt_1 \quad (4)$$

It is obviously that the integral computation becomes more complicated if the given path contains more interchange stations, since more travel time combinations can be generated.

Similarly, to ease the computation of integral in Eq. 4, we first let the travel time be considered in the discrete manner, then the integral can be degraded to the sum computation. Taking the path ($path_1 : s_o \rightarrow s_1 \rightarrow s_d$) shown in Fig. 1 as an example, its arriving-on-time probability within 15 minutes can be computed as:

$$P_{ij}(t \leq 15|path_1) = \underbrace{0.3 \times (0.6 + 0.4)}_{\text{Case I}} + \underbrace{0.7 \times 0.6}_{\text{Case II}} = 0.72$$

For the given path, two cases in total can lead to the successful arrival of packages by the deadline. **Case I:** If the first recruited taxi driver spent no more than 5 minutes in the first segment (i.e., $s_o \rightarrow s_1$), due to the sufficient time margin left, then the second recruited taxi driver can arrive at s_d by the deadline *at a hundred percent*. **Case II:** If the first recruited taxi driver took more than 5 minutes in the first segment, then the package can be arrived on time *only if* the second recruited taxi driver spent no more than 5 minutes to accomplish the second segment (i.e., $s_1 \rightarrow s_d$).

Theorem 1. *For a unique path, its arriving-on-time probability becomes higher (or at least unchanged) if allowing a longer deliver time, mathematically, we have:*

$$P_{ij}(t \leq t_1|Path) \leq P_{ij}(t \leq t_2|Path), \ if \ t_1 < t_2 \quad (5)$$

Proof: The theorem can be proven by induction, detailed as follows:

Base Case: When $n = 1$ (n is length of the given path which is quantified by the number of interchange stations contained by the path minus one), it is obviously that we have $P_{ij}(t \leq t_1|Path^{n=1}) \leq P_{ij}(t \leq t_2|Path^{n=1}), \ if \ t_1 < t_2$, according to *Definition* 6.

Induction Step: Let $k \in N$ be given and suppose Eq. 5 is true for $n = k$, that is, we have:

$$P(t_2|Path^{n=k}) \geq P(t_1|Path^{n=k}), \ if \ t_1 < t_2$$

Then,

$$P(t_1|Path^{n=k+1}) = \int_0^{t_1} p(t)P(t_1 - t|Path^{n=k})dt$$

where $p(t)$ is the probability density function on the edge from the origin to the first stop.

$$\begin{aligned} P(t_2|Path^{n=k+1}) &= \int_0^{t_2} p(t)P(t_2 - t|Path^{n=k})dt \\ &= \int_0^{t_1} p(t)P(t_2 - t|Path^{n=k})dt + \\ &\quad \int_{t_1}^{t_2} p(t)P(t_2 - t|Path^{n=k})dt \\ &\geq \int_0^{t_1} p(t)P(t_1 - t|Path^{n=k})dt + \\ &\quad \int_{t_1}^{t_2} p(t)P(t_2 - t|Path^{n=k})dt \\ &= P(t_1|Path^{n=k+1}) + \int_{t_1}^{t_2} p(t)P(t_2 - t|Path^{n=k})dt \\ &\geq P(t_1|Path^{n=k+1}) \end{aligned}$$

Conclusion: By the principle of induction, Eq. 5 is true for all $n \in N$. ■

Definition 9 (Maximum Probability of Arriving-on-Time)**.** *For an OD pair, the maximum probability of arriving-on-time (with time cost no greater than t_0) is defined as the maximal one among all probabilities on all possible N paths from the origin (s_i) to the destination (s_j), denoted by $u_{ij}(t \leq t_0)$. In another word, $u_{ij}(t \leq t_0)$ serves as the upper bound of $P_{ij}(t \leq t_0)$.*

If a package at node s_i is sent to s_k in the next step, the probability that the package spends a time in the interval $[\omega, \omega + d\omega]$ on edge s_i, s_k is $p_{ik}(\omega)d\omega$, thus the time margin at node s_k is $t_0 - w$. On the basis of *Bellman's principle of optimality* [5], [26], no matter which node s_j that the package is elected to send next, the package must follow the optimal routing strategy in shipping from node s_k to the destination s_j within the remaining time $t_0 - \omega$. Therefore, the maximum probability of arriving-on-time can be formally defined *recursively* as follows:

$$u_{ij}(t \leq t_0) = \max_{k \neq j} \int_0^{t_0} p_{ik}(\omega)u_{kj}(t \leq t_0 - \omega)d\omega \quad (6)$$

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

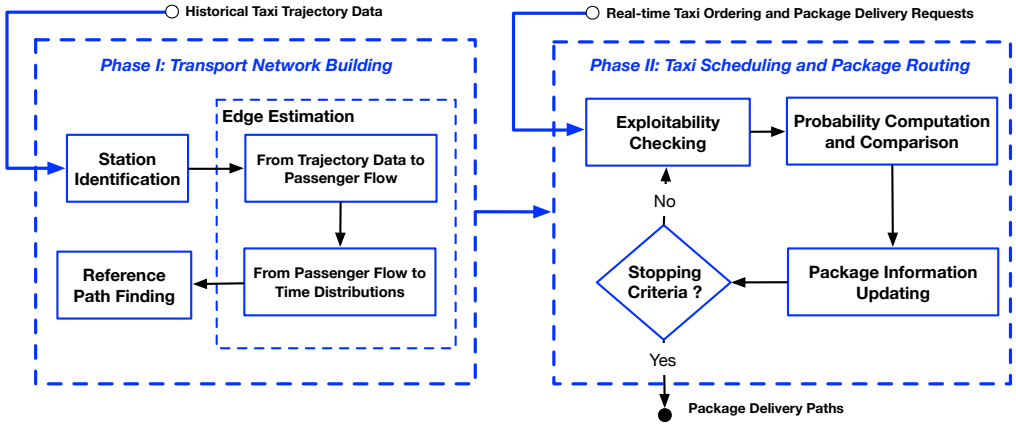


Fig. 2. The overview of the CrowdExpress system.

Intuitively, according to *Definition* 6, to compute the maximum probability of arriving-on-time for an OD pair, *one needs to find all possible paths from the origin to the destination*, which is well-known as an NP-hard problem [18]. To make this point clear, we use the example shown in Fig. 1 again. It is easy to find all paths from s_o to s_d in this example, i.e., $path_1 : s_o \rightarrow s_1 \rightarrow s_d$, $path_2 : s_o \rightarrow s_2 \rightarrow s_d$, $path_3 : s_o \rightarrow s_3 \rightarrow s_d$ and $path_4 : s_o \rightarrow s_4 \rightarrow s_3 \rightarrow s_d$, respectively. For each path, similar to the computation in Eq. 4, it is not difficult to obtain the corresponding arriving-on-time probability within a given deadline (e.g., $t_0 = 15$ minutes). As a result, $u_{ij}(t \leq 15) = 0.95$ for the example when delivering the package via $path_2$. It is obvious that $u_{ii}(t) = 1$ given any deadline, since no travel time is needed if the package stays still. We exclude the round trip in the study.

2) Assumptions: In this work, we make the exactly same assumptions to [9], regarding taxi driver participation constraints and willingness, as well as the package traceability. To avoid overlapping and save room, we do not elaborate the details here. Readers can refer to [9] for the assumption details.

B. Problem Statement

The collaborative crowdsourced package deliveries leveraging the relays of passenger-occupied taxis can be viewed as the problem of finding arriving-on-time paths, and thus can be formulated as follows:

Given:

1) A historical set of taxi trajectory records $\{Tr\}$, such as from the past month in the designated city,
2) A set of real-time taxi ordering requests TOR from mobile phone apps, and a set of real-time package delivery requests PR. Note that these two requests come in stream,
3) A given deadline specified by the user for each package delivery request.

Objectives:

1) Build a package transport network (i.e., the identification of interchange stations and estimation of edge values) based on the historical taxi trajectory data.
2) Find a package delivery path for each package request (pr), which can make the package arrive at the destination by the deadline. However, such package delivery path may not be unique or existed. To migrate the issue, *we thus transform the problem to the arriving-on-time problem, i.e., finding the optimal one that is expected to have the maximum probability of arriving-on-time*[3].

Constraints:

1) Only taxis that accept the real-time taxi ordering requests after the package delivery request is posted can be scheduled, i.e. $\{tor.r_t\} > pr.t_p$.
2) A recruited taxi can be available to participate again only after completing the current task (i.e., dropping off the package at the predefined interchange station).

C. System Overview

We develop a two-phase system called CrowdExpress, i.e., *offline package transport network building* and *online taxi scheduling and package routing* to find the optimal route with the maximum probability of arriving-on-time for each package delivery request within a given deadline, by collaboratively recruiting taxi drivers that have been reserved to passengers (occupied by passengers), as shown in Fig. 2.

Phase I is an *offline* process, with the historical taxi trajectory data as input, aiming to identify the package interchange stations, estimate the edge values, as well as find the reference paths for any given OD package pairs. Based on the constructed package transport network, for a real-time incoming individual package delivery request, Phase II mainly takes four *online* steps to tackle with, namely, *Exploitability Checking, Probability Computation and Comparison, Package Information Updating and Stopping Criteria Checking*, with the streaming taxi ordering requests as input. The system finally outputs the corresponding package delivery paths. The technical details will be presented in the next two sections.

[3]If the optimal one is still unable to send the package on-time, then it is safe to claim that the package delivery is an unsuccessful one.

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

IV. Phase I: Offline Package Transport Network Building

The task of offline package transport network building is to identify interchange stations (i.e., node locations) as well as the estimation of travel time distributions (i.e., edge values). Here, we mainly take a three-step procedure to achieve the objectives, detailed as follows.

A. Package Interchange Station Identification

One basic principle for the identification of interchange stations is that they should be located where passengers are frequently picked-up or dropped-off, to take as full advantage of passenger-sending rides for the package hitchhiking as possible, and probably to minimize the extra efforts imposed to the taxi drivers as well. Fortunately, with the taxi trajectory data left, such information (i.e., pick-up and drop-off points) can be easily extracted. Then, we cluster them using *DBSCAN* algorithm as it is capable of merging closer data points with arbitrary distributions [16]. Finally, locations near the point centroids of each cluster and the road sides are identified as the locations of the interchange stations to serve the purposes of package collection, storage and receiving.

B. Edge Estimation

1) From Trajectory Data to Passenger Flow: It is straightforward to infer the passenger flow between any two interchange stations during a given time slot from the trajectory data [9]. Specifically, we *first* group the trajectories according to their starting time (t_s). *Second*, to compute the passenger flow from s_i to s_j, we count the number of the trajectories satisfying Eqns. 7∼8. It should be noted that there could be no passenger flow between some interchange station pairs.

$$Ddist(Tr_i.o, loc(s_i)) \leq \epsilon \tag{7}$$

$$Ddist(Tr_i.d, loc(s_j)) \leq \epsilon \tag{8}$$

where $Tr_i.o$ and $Tr_i.d$ are the original and destination points of Tr_i, respectively; $loc(\cdot)$ gets the latitude and longitude location of the given interchange station; $Ddist(a{\cdot}b)$ calculates the *driving distance* from point a to point b; ϵ is a user-specified parameter. The physical meaning of ϵ is that any passenger-delivery ride which starts and ends near a pair of interchange stations (i.e., with driving distance less than ϵ) can be hitchhiked for the package delivery between this pair. Hence, for a given OD pair, a bigger ϵ would result in a bigger number of passenger flow. It is worth noting that, for a specific trajectory, there could be multiple interchange station pairs that satisfy Eqns 7∼8, in other words, can provide package hitchhiking ride between all these pairs. Therefore, a bigger ϵ also leads to a bigger number of interchange station pairs, suggesting that the corresponding trajectory can be more capable of providing hitchhiking rides. However, for passengers, a bigger ϵ may mean a longer waiting time for the reserved taxis, since the taxi driver might have to travel farther to collect the package before picking up passengers. To control for the additional waiting time, we set ϵ to 500 meters.

2) From Passenger Flow to Time Distributions: To estimate an edge value, we need to estimate two parts, i.e. the *waiting time* and the *driving time* [9]. The *driving time* is simply the travel time of each taxi trajectory.

The *waiting time* on the edge is defined as the time required to wait for a suitable hitchhiking ride that can transport a package from s_i to s_j directly. To address this problem, we employ the *Non-Homogeneous Poisson Process* (NHPP) to model the behavior of passenger taking taxis [35]. According to the statistical *frequency* of passenger taking taxis from s_i to s_j in history (i.e. passenger flow), we can estimate the *waiting time* of packages at different time slots at the interchange stations. Specifically, the *waiting time* on the edge from s_i to s_j is:

$$t_w = waiting\ time = \frac{\Delta T}{\bar{N}} \tag{9}$$

where $\bar{N}$ is the average number of passengers taking taxis from s_i to s_j during the given time slots; ΔT is the length of that time slot. Note that the *waiting time* obtained by Eq. 9 is in the *statistical sense*, and it could be much smaller in the real case due to the timely *availability* of right passenger-sending trips.

The waiting time component is substituted in advance when computing the arriving-on-time probability of a given path at a given time duration, thus the corresponding *time distributions* is obtained by discretizing the driving time component *only* according to *Definition* 7.

C. Reference Path Finding

On the basis of the constructed package transport network, we find two reference paths for each given OD pair, i.e., the shortest path when assuming value on each edge is the minimum travel time ($SPath\_min$), and the shortest one when assuming value on each edge is the maximum travel time ($SPath\_max$), respectively. More specifically, given an OD pair, when choosing all minimum travel times on edges, we can find the shortest path, i.e. $SPath\_min$, using *Dijkstra's* algorithm [31]. $SPath\_min$ refers to the case that the package can be delivered in the most efficient manner if sent via that path. Similarly, when choosing all maximum travel times on edges, we can also discover the shortest path, i.e. $SPath\_max$. $SPath\_max$, as a comparison, refers to the case that the package can be delivered in the lowest efficient manner if sent via that path. In addition, the corresponding total travel times of those two paths are also recorded, which can be used to guide the online taxi scheduling and package routing upon the real-time taxi ordering requests in the second phase, with details would be further addressed in the next section. It should be noted that, although it is a time-consuming procedure to find two shortest paths for each OD pair with a computation complexity of $\mathcal{O}(k^2)$, it can be operated offline[4].

[4]Suppose there are k nodes in the network, the maximum number of all OD pairs is $k(k-1)$.

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

V. Phase II: Online Taxi Scheduling and Package Routing

Given an OD pair of the package, the task of *online taxi scheduling* is essentially a sequential decision-making one. To be more specific, the phase should make a decision whether utilizing the *current* available hitchhiking ride for the package delivery or waiting for the *future* rides, according to the upcoming taxi ordering requests generated in real-time and the remaining time margin. The package would be hitchhiked immediately if the corresponding maximum arriving-on-time probability is greater than the expected value if waiting for the future opportunities. The upper bound value of maximum arriving-on-time probability if hitchhiking the current ride is actually related to *the case of sending the package via the highest efficient path to the final destination after reaching the intermediate stop (same to the destination of current ride) with the help of current ride.* Refer to Section V-B1 for the details. While the expected maximum arriving-on-time probability if hitchhiking the future rides corresponds *the case of sending to the other intermediate stops before sending the package to the final destination from that intermediate stop via the most efficient path.* Refer to Section V-B2 for the details. We thus propose **maxProb** algorithm to compute and compare these two cases iteratively until reaching the package's destination to determine whether the waiting is worthy.

While the task of *online package routing* is much simpler, i.e., just assigning the delivery task to the scheduled taxi. As a result, the package will be delivered to the next stop, which is same to the destination of the taxi. The two steps impact each other *mutually*. On one hand, both the potential hitchhiking rides for the package delivery and the time margin is highly related to the current stop where the package locates; on the other hand, which the next stop that the package will head to is determined by the scheduled taxi. In the following, we mainly focus on the step of online taxi scheduling, which includes the following operations.

A. Exploitability Checking

Before triggering this phase, we first need to conduct the *exploitability checking*, i.e., to determine: 1) whether the origin of an upcoming taxi ordering request is close to the location of the package; and 2) whether it ends at one of the interchange stations. If both conditions are met, then we further need to compare the maximum arriving-on-time probability of sending the package via s_k (*hitchhiking the current*) to the maximum arriving-on-time probability of sending the package via other potential next stations (the other neighbours of s_o in the transport network) (*waiting for the future*). If the former value is greater, then the package will be sent out *immediately* by hitchhiking the *current* taxi ride; otherwise, the system will *wait* for the new *future* taxi ordering requests that may lead to a higher arriving-on-time probability and the decision will made again, given the new time margin. Thus, the core of the online taxi scheduling is the maximum arriving-on-time probabilities *computation and comparison*.

B. Probability Computation and Comparison

1) Probability Computation if Hitchhiking the Current:

According to the definition, the maximum arriving-on-time probability for case *if hitchhiking the current* (suppose that the recruited taxi will go to s_k) can be computed as follows:

$$P_{od}(t \leq t_0 - \Delta t | Path_{okd}) = \int_0^{t_0-\Delta t} P_{ok}(\omega) u_{kd}(t \leq t_0 - \Delta t) d\omega \tag{10}$$

where $Path_{okd}$ refers to the path from s_o to s_d while stopping at s_k in the next; Δt is the time difference between the occurring time of the taxi ordering request and the birth time of the package delivery request, i.e. $\Delta t = tor.t - pr.t$; t_0 is the given time duration of the deadline; u_{kd} is the maximum arriving-on-time probability from s_k to s_d, as defined in *Definition* 6.

By definition, it is easy to compute the arriving-on-time probability of a determined path under a given deadline time, such as $\int_0^{t_0-\Delta t} P_{ok}(t)dt$. By contrast, it is rather challenging to get the value of the latter part in Eq. 10. As discussed, one naive and straightforward way is *first* to enumerate all the possible paths from s_k to s_d, *then* compute the value of arriving-on-time probability for each of them, *finally* pick up the maximum one as the final value. It is easy to understand that the trivial method cannot work in real cases as the problem of finding all possible path for a given OD pair is NP-hard. Actually, it is also no need and some branches in the transport network can be *trimmed recursively*. We propose a novel algorithm named **maxProb** to compute the probability, which mainly consists of two operations, i.e, initialization and deep-first searching.

Initialization: From s_k to s_d, it will be easy to find two shortest paths, $SPath\_min(s_k, s_d)$, $SPath\_max(s_k, s_d)$. We further obtain the two corresponding reference paths from s_o to s_d via s_k, and compute their arriving-on-time probabilities given the remaining time, which are two boundaries and used to guide the process of *branch trimming*. For brevity, we use $minP_{o\to k\rightsquigarrow d}$ and $maxP_{o\to k\rightsquigarrow d}$[5] to represent the arriving-on-time probabilities of $s_o \to SPath\_min(s_k, s_d)$ and $s_o \to SPath\_max(s_k, s_d)$, respectively.

Depth-First-Searching: From s_k to s_d, we mainly apply the *Depth-First-Search* (*DFS*) method to recursively get each possible path [33], and compute the maximum probability of arriving-on-time. One exception is that the user specifies an extremely long deadline, mathematically, $maxP_{o\to k\rightsquigarrow d} = 1$, implying that the package can be delivered on time for sure via the reference path. Therefore, no *DFS* is needed and a simple taxi scheduling can be enough under such circumstance. The overall procedure of *DFS* starting from s_k can be summarized as follows.

The core function is to find the next package stop of s_k, with the pseudocode shown in Algorithm 1. The very beginning task is to get the neighouring stations of s_k, given the topology of

[5] $a \to b$ refers to the package is sent from s_a to s_b directly, while $a \rightsquigarrow b$ refers to the package is sent from s_a to s_b via some intermediate stops that are determined by the reference paths between s_a and s_b.

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 $FindNextStop(s_k, s_d, TN)$

1: $ngb = Neigh(s_k, TN)$; //get the neighbouring stations of s_k based on the transport network topology.
2: $ns = \emptyset$;
3: $refP = minP_{o \to k \rightsquigarrow d}$;
4: **for** $i == 1$ to $|ngb|$ **do**
5: $s_{ki} = ngb(i)$;
6: **if** $refP \leq P(t \leq t_0 - \Delta t - t\_min(s_{ki}, s_d) | Path_{o \to k \to ki})$ **then**
7: $ns = ns \cup ngb(i)$;
8: **end if**
9: **end for**

the built package transport network (Line 1). *DFS* starts to find the next stop from one of the neighbouring nodes of s_k (e.g., s_{ki}) in the loop (Lines 4∼9). Whether s_{ki} can be the next package stop is determined by the inequation shown in Line 6. In the in-equation, the reference probability ($refP$) is first set to $minP_{o \to k \rightsquigarrow d}$ (Line 3). $t\_min(s_{ki}, s_d)$ is the time cost of the reference path $SPath\_min(s_{ki}, s_d)$ estimated by the historical taxi trajectory data; the right part of the in-equation is the maximum arriving-on-time probability which corresponds to the *ideal* case that the package can be shipped from s_{ki} to s_d in the most-efficient way (i.e., the time cost on each edge in the package transport network is the minimal). If the in-equation satisfies, *it indicates that there exists a potential path which can lead to a higher maximum arriving-on-time probability than the reference path*, thus *DFS* will continue to search with a new start from s_{ki} *recursively*, with the same procedure to the *DFS* starting from s_k; otherwise, *DFS* will be terminated and the related branches will be trimmed at the same time. Thus, a recursion may be stopped either at some intermediate node or generates a successful path reaching the given destination. If a *valid* path is resulted ($Path\_valid$), $refP$ will be updated using its corresponding probability *if and only if* it is greater than the previous value.

The whole *DFS* ends when all neighhours of s_k are checked by *repeatedly* calling the above recursive *DFS*. Finally, the maximum arriving-on-time probability if assigning the package delivery task to the current available taxi shall be the final value of $refP$.

2) Probability Computation if Waiting for the Future: If the future taxi rides heading to any one of the other neighouring nodes of s_o except for s_k (marked as $\{ngb(s_o)\} - s_k$) could lead to a higher arriving-on-time probability, compared to the case if *hitchhiking the current*, a better decision should be the *waiting*. The maximum arriving-on-time probability *if waiting for the future* can be computed as follows:

$$P_{od}(t \leq t_0 - \Delta t' | Path_{ojd}) = \max_{s_j \in \{ngb(s_o)\} - s_k} \int_0^{t_0 - \Delta t'} P_{oj}(\omega) u_{jd}(t \leq t_0 - \Delta t') d\omega \quad (11)$$

where $\Delta t' = \Delta t + t_w(s_o, s_j)$ and $t_w(s_o, s_j)$ refers to the edge value component of waiting time from s_o to s_j. As can be seen, the major difference between Eq. 10 and Eq. 11 is the time margin. More specifically, less time margin is left for the package deliveries as an additional time cost would be induced while waiting for the future taxi rides. Here, we simply use the average waiting time to approximate the additional time cost.

Similarly, all maximum arriving-on-time probabilities of waiting for the future exploitable taxi rides from s_o can be computed, and the maximal one among them will be chosen to represent the maximum arriving-on-time probability if assigning the package delivery task to the future taxis.

3) Probability Comparison: As discussed, once receiving a real-time taxi ordering request, *on-line taxi scheduling and package routing* will be activated, and the package may be shipped to some intermediate stop by hitchhiking the current ride or stands still at the current stop by comparing those two maximum arriving-on-time probabilities. Note that the remaining time margin shrinks as time goes by, the two probabilities computed in Eqns. 10 and 11 are *dynamically changed*, thus the better decision (hitchhiking the current or waiting for the future) can be also adjusted *adaptively* "on-the-fly".

C. Package Information Updating

After the package is sent to the next station whether by hitchhiking the current or future rides, the information about the package delivery request should be updated. To be more specific, the origin of the package should be set as the updated station that the package locates; the birth time should be set as the time when the package arrives at the current stop. The newly updated package delivery request will be used as the input of Phase II.

D. Stopping Criteria Check

For a package delivery, the previous three operations will be *iteratively* conducted until one of the following two stopping criteria is satisfied: 1) the package has arrived at its destination; 2) the time is running out (the package cannot be delivered by the deadline), in that case, the system would report *failure*. For those failure package deliveries, empty taxis can be recruited dedicatedly to send them to the destinations. However, the topic is out of the scope of the paper.

VI. Experimental Evaluations

In this section, we empirically evaluate the performance of the proposed **maxProb** algorithm. We first introduce the experimental setup, baseline algorithms used for comparison, evaluation metrics and results on algorithm efficiency and effectiveness. We discuss some open research issues to be further addressed in the end.

A. Experimental Setup

1) Experimental Data: We use the real-world datasets for the evaluation, i.e. the road network data which is extracted from OpenStreetMap[6], and one month of taxi trajectory data

[6]www.openstreetmap.org

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

TABLE II. STATISTICS OF THE TAXI TRAJECTORY AND ROAD NETWORK DATA SETS.

Datasets	Properties	Statistics
Taxi trajectory	Number of taxis	>19,000
	Number of occupied rides	≈ 13 M
Road network	Number of road intersections	11,999
	Number of road segments	15,202

generated by over 19,000 taxis in the city of New York (NYC), US. Readers can refer to [1] for the details on how to extract the road network from the crowd-sourced open platform (i.e., OpenStreetMap) correctly. We determine package interchange stations according to the algorithm discussed earlier.

For the taxi trajectory data, we split it into training and testing sets, according to the date of the month. Specifically, the training set contains taxi trajectories on 1st∼20th, January, 2013, which are used to build package transport network. It should be noted that for the taxi trajectory data in NYC, no detailed travel routes between the pick-up and drop-off points are provided due to the privacy considerations. The testing trajectories were generated from 21st to 31st, January, 2013, which are used as the real-time taxi ordering requests (TOR) for testing the performance of the proposed **maxProb** algorithm. Table II shows some basic statistics of the taxi trajectory and road network data. It should be noted that we do not differentiate the trajectory data collected from work days and rest days. We simply exclude trajectory data collected in rush hours in work days since the passenger flow pattern is quite different at that time and the operation of CrowdExpress at that time may worsen the traffic conditions.

2) Package Delivery Request: Since the data sets do not contain information about package delivery requests, we apply simple mechanism to simulate it. The novel package delivery system targets the city-wide person-to-person service. Hence, to simulate a package delivery request (PDR), we randomly generate its birth time, origin and destination. Regarding the origin and destination, any package should be originated and ended at the interchange stations. We further eliminate requests with short-distance OD pairs (i.e., with driving distance less than 3 km) since few users would request speedy shipping as ordinary delivery may be equally efficient in this case. Besides, based on the total number of pick-ups and drop-offs of taxi passengers in the station, we categorize the stations into two kinds, i.e., the popular and the unpopular.

3) Evaluation Environment: All the evaluations in the paper are programmed using Java language under the Eclipse J2SE 1.5 integrated development environment, and run on an Intel Core i5-4950 PC with 8-GB RAM and Windows 7 operation system.

4) Evaluation Metrics: We adopt the following three metrics to evaluate the proposed **maxProb**.

Success Rate. The success rate is the ratio of the number of packages which can be delivered successfully within a given deadline (i.e. time duration) to the number of total packages (i.e. the number of package delivery requests simulated).

$$SR(t \leq deadT) = \frac{|\mathcal{TC}(Path(o_p \rightsquigarrow d_p)) \leq deadT|}{|PQ|} \quad (12)$$

where $Path(o_p \rightsquigarrow d_p)$ represents the optimal path generated by the proposed **maxProb** algorithm for a given package delivery request; $deadT$ is the given deadline. The delivery performance is better if the success rate is higher within a given shorter deadline.

Regarding the *deadline setting*, we do not set an *absolute* value for all package deliveries since package originated (ended) at different locations would need *absolutely* varied time. Thus, for an individual package delivery, we set a *relative* deadline separately instead, according to the following equation:

$$deadT = t_{avg} + extraT \quad (13)$$

in which t_{avg} is the average value of the time cost by the two reference paths, which is obviously different for packages with different OD pairs. $extraT$ is the extra time value imposed by the user; a smaller $extraT$ indicates that the user needs the package more urgently and wants it to be arrived more timely.

Number of Relays. The number of relays (Num_{relays}) during a package delivery is defined as the number of participating taxis Num_{p_taxis} (Formula 14).

$$Num_{relays} = Num_{p\_taxis} \quad (14)$$

On one hand, fewer relays generally mean a lower chance of package loss or damage, and perhaps less overhead cost. On the other hand, fewer number of participating taxis may imply requiring less reward cost to taxi drivers. Thus, the performance is better if the number of relays is smaller.

Package Throughput. The average number of package deliveries that the system can complete successfully per day. The system achieves better performance if the package throughput is bigger.

B. Baseline Algorithms

To show the superior performance of our proposed algorithm, we compare it with the following three baseline algorithms.

(1) **FCFS** - This method adopts the *First Come First Service* strategy. Specifically, the package will be assigned to the *first* taxi that will pick up a passenger near the interchange station that the package locates, *regardless of its destination*. In fact, this algorithm always favors the strategy of *hitchhiking the current*, which is also known as an extension of the simple and well-known *flooding* strategy.

(2) **DesCloser** - This method assigns the package to the *first* taxi that will head to somewhere closer to the destination of the package, compared to the current station of the package. This algorithm implements a distance-based geo-cast scheme that is commonly seen in other domains.

(3) **Direct** - This method waits for the taxi heading to the destination near the interchange stations that the package will be delivered *directly*, without any intermediate stops. Specifically, the package will be assigned to the taxi that will pick up and drop off a passenger near the interchange stations that the package locates and heads, respectively. Thus, no relays are needed.

Remark. Each relay in **DesCloser** is effective as it ensures that the package would move towards its destination step by

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

step; while some relays in **FCFS** can be ineffective as the package moves further away from its destination. For **Direct**, it may be inefficient for package deliveries where there is little passenger flow in-between. However, all three baseline algorithms do not take the arriving-on-time probability of package deliveries into account, thus probably resulting in a high failure rate.

C. Experimental Objectives

We plan the experiments to address the following questions.

Question 1: How much computational resource is required to generate the response for a package delivery request?

Question 2: How does **maxProb** *perform under different given deadlines?*

Question 3: How does **maxProb** *perform w.r.t the birth time of package deliveries?*

Question 4: How does **maxProb** *perform w.r.t the locations (both origins and destinations) of package deliveries?*

Question 5: How many packages can be delivered daily on average (i.e., throughput) with the proposed system?

The first question concerns the efficiency of **maxProb**, and *Questions* 2∼4 are related to its effectiveness. To answer the first question, we compute the *response time* of the algorithms. Since passenger flows are both time- and space-dependent, to assess the effectiveness of the different algorithms, we calculate their success rates and the number of relays with respect to packages to be dispatched to different parts of the city at different time of the day. We test the throughput of the proposed system and the success rate under different number of package requests generated per hour, and also examine the system throughput given different number (density) of interchange stations in the designate city (*Question 5*).

D. Experimental Results

1) Results of Response Time: We first analyze the main *operations* involved in the four algorithms respectively. For a given package delivery request (pr), when a new real-time taxi ordering request (tor) comes in, all four algorithms need to determine whether tor starts near the origin of the package and stops at some interchange station, i.e., *exploitability checking*. **FCFS** will recruit the first taxi that satisfies the criteria, but for **DesCloser**, it needs to further determine whether the heading destination of the taxi is closer to the destination of the package, compared to its current location. For **Direct**, it also needs to determine whether the taxi would head to the destination of the package. Thus one more *comparison* operation is required for both **DesCloser** and **Direct** algorithms. For **maxProb**, the procedure is even more complicated, mainly requiring additional *probability computation and comparison* operations, as discussed previously. Each algorithm needs to repeat its own operation procedure at each intermediate station (except for **Direct**) and thus the total response time is the *accumulated* computational time over all iterations.

We show the comparison results of average response time of the four different algorithms in Fig. 3. The average response time of **FCFS** is the biggest while that of **Direct** is the

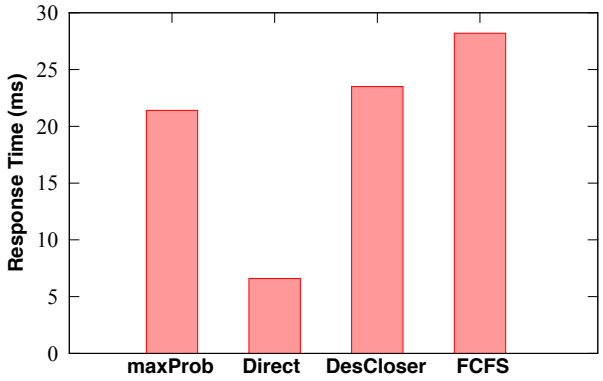


Fig. 3. Evaluation results of response time for four different algorithms.

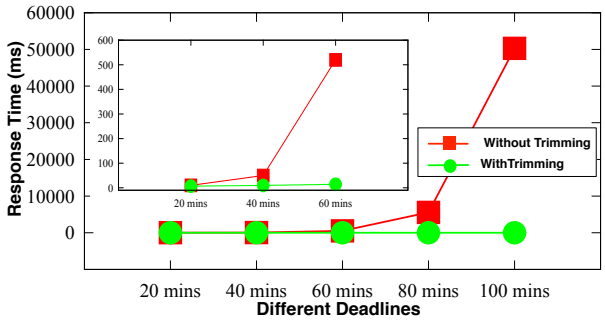


Fig. 4. Comparison results of response time of **maxProb** with/without branch trimming w.r.t different deadlines.

smallest among all algorithms. The average response times of **DesCloser** and **maxProb** are in-between and **DesCloser** costs slightly more time than **maxProb**. More precisely, the average response time of **Direct** is within 7 milliseconds; the average response time of **maxProb** is around 22 milliseconds; the average response time of **FCFS** is no more than 30 milliseconds. All results indicate that all four algorithms are quite efficient, and can plan and adjust the shipping routes in real time. We also observe an interesting phenomenon: **FCFS** only involves two simple comparisons for each candidate taxi, but it is the most time-consuming method accumulatively. In comparison, although **maxProb** algorithm contains the most sophisticated operations (i.e., *DFS*), it needs a shorter response time than **FCFS** and **DesCloser**. We argue that this is because: **FCFS** and **DesCloser** require more rounds of computation (i.e., more relays) than the **maxProb** algorithm. In another word, both **FCFS** and **DesCloser** generate more unnecessary package relays. The efficiency of **maxProb** can be guaranteed because: 1) the number of connected intermediate stops to the package's origin is limited, and 2) all the paths with the highest efficiency between any two stops in the transport network are pre-computed and stored.

We further evaluate the effectiveness of the branch trimming in the probability computation for **maxProb** in terms of the average response time saving, with the result shown in Fig. 4. To better illustration, we also highlight the result w.r.t different deadlines within the range of [20, 60] minutes in the left-

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

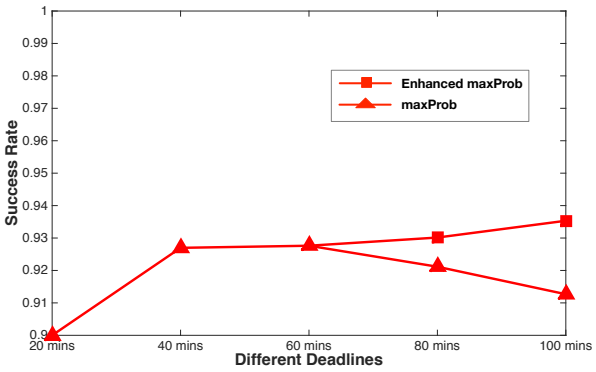


Fig. 5. Evaluation results of success rate under different deadlines.

top part of the figure. A significant time saving is obtained with the introduction of branch trimming. To be more specific, the gap of the average response time with/without branch trimming increases exponentially wider as the given deadline becomes bigger, what is more, all the average response times with branch trimming remain stable and small under all given deadlines. The package delivery requests are the same for the efficiency studies, with a number of 100.

2) Results of Success Rate: We present results of the performance of **maxProb** in terms of the success rate under different deadlines in Fig. 5. More specifically, $extraT$ is set in a range from 20 to 100 minutes, with an equal interval of 20 minutes. As one can see, the success rate under all deadlines is high, with a value above 90%. The success rate firstly becomes slightly higher then decreases gradually as users allow a longer deadline. The highest success rate appears when the $extraT$ is set 60 minutes, with the value of around 93%. The observed phenomena seems *somehow counterfactual* at the first glance as the success rate should be higher when setting a longer deadline in intuition. The root cause is that: when giving a bigger deadline, the arriving-on-time probability if hitching the current becomes greater, as guaranteed by Theorem 1. An extreme case is that the probability would always equal one and *dominates* the other possibilities, as a consequence, the **maxProb** algorithm tends to select the *hitchhiking the current* strategy while routing packages. Under such circumstance, **maxProb** degrades to the **FCFS** algorithm to some extent, causing the negligible decrease of the success rate. To avoid such degradation, the key issue is to lower the value computed by Eq. 10. Thus, one potential solution can be the reduction of remaining time margin during package routing when applying **maxProb** algorithm. Specifically, if *hitchhiking the current* strategy is always preferred at first during package routing, the remaining time margin can be manually reduced to 90% of the true one that imposed by the user. We call such variant as the enhanced **maxProb** algorithm (i.e., introducing the modification of the time margin). The effectiveness of the enhanced **maxProb** algorithm is evaluated, with the results shown in Fig. 5. With the enhanced version, the success rate increases when the deadline gets longer. It should be noted that the success rate of **FCFS** is much lower, compared to

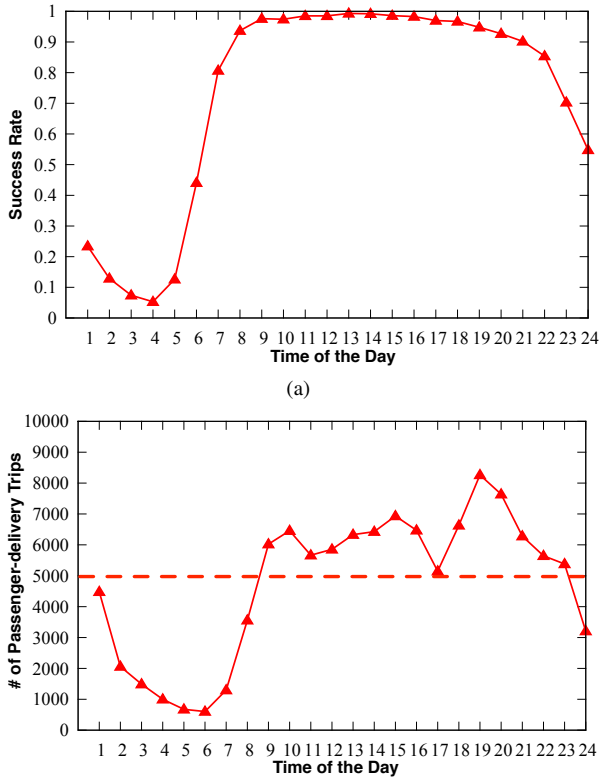


Fig. 6. Evaluation results of success rate under different birth times of the package deliveries.

maxProb, which will be verified in the following experiments. The number of package delivery requests is 10,000, with the birth time uniformly distributed at the day-time (i.e., from 8:00 to 18:00).

We are also interested at the performance of the success rate at different time of the day. As can be observed in Fig. 6(a), the success rate is quite *stable and high* from 8:00 to 18:00, with the value above 95%, demonstrating the usefulness of the proposed system in practice during the day time. The success rate is extremely low during late-night and early-morning. For instance, the success rate is even lower than 10% from 2:00 to 5:00. To get an in-depth understanding on the root cause, we further show the number of passenger-delivery trips at different time of the day in Fig. 6(b). As expected, the success rate is highly correlated with the number of passenger-delivery trips, i.e., a greater number of passenger-delivery trips implies a higher success rate since more hitchhiking opportunities for package deliveries are provided. An interesting observation is that the success rate at 17:00 is still high though the number of passenger-delivery trips is not large at that time, compared to nearby hours. This is because: on one hand, the number of hitchhiking opportunities for package deliveries should be accumulated during the given deadline (usually bigger than an hour); on the other hand, the number of passenger-delivery trips in the next two hours increases and remains high. On the contrary, the number of passenger-delivery trips during late-

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

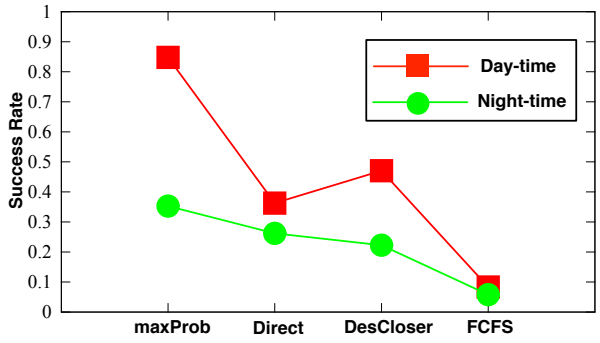


Fig. 7. Evaluation results of success rate for four different algorithms at day time and night time respectively.

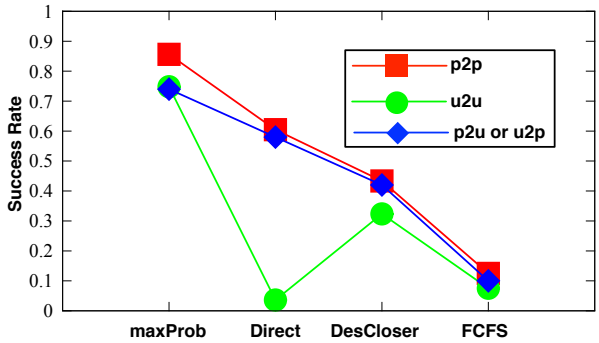


Fig. 8. Evaluation results of success rate for four different algorithms of different package categories.

night and early morning is *continuously* small. During that time periods, although hitchhiking opportunities are accumulated, it is still *insufficient*, resulting in a poor success rate. In this study, the number of package delivery requests is 1,000 for each hour time; the $extraT$ is fixed to 60 minutes.

We report the result of success rate for four different algorithms w.r.t the time of the day. For simplicity, we do not provide the success rate of the four algorithms under every hour of a day, and just split a day into two time slots, i.e, day time from 8:00 to 18:00 and the rest is the night time, respectively. In the time dimension, as shown in Fig. 7, for all four algorithms, the success rate is higher at day time than night time, except for **FCFS** which only achieves the similarly low success rate at both time slots. **Direct** algorithm returns quite close performance at day time and night time. As predicted, the success rate of **FCFS** algorithm is the lowest, i.e., below 10%. Compared to the other three baseline algorithms, **maxProb** achieves the best success rate at both day time and night time. In this experiment, the number of package delivery requests is 10,000, with the birth time uniformly distributed at the day time and nigh time, respectively; the $extraT$ is fixed to 60 minutes.

In real situation, there are more passengers who prefer to take taxis at some interchange stations, providing more hitchhiking opportunities for package deliveries, probably leading to a better success rate. We thus manually categorize the interchange stations into two classes (i.e. *popular, unpopular*) in advance, by taking its total number of pick-ups and drop-offs in history into account. The number of interchange stations labeled as *popular* is 19; and the number of interchange stations labeled as *unpopular* is 15. We further identify three categories of package delivery requests, according to the labels of the original and destination station, as shown in Table III. Then, we test the performance on success rate for each category. The number of package delivery requests for each category is same, with a value of 10,000.

Fig. 8 shows the comparison of success rate of the four algorithms under a given deadline (the $extraT$ is fixed to 60 minutes in this study) for the three categories. **maxProb** achieves the best performance for three categories, while the performance of **FCFS** is the worst. One exception is the success rate of **Direct** for the Category III packages. In such case,

TABLE III. THREE IDENTIFIED CATEGORIES OF PACKAGE DELIVERY REQUESTS.

	Description
Category I	*packages start and end at popular stations* (p2p)
Category II	*packages start or end at popular stations* (p2u or u2p)
Category III	*packages start and end at unpopular stations* (u2u)

without any relay at the intermediate stations, **Direct** obtains an extremely low success rate (i.e., under 5%), which is due to the fact that there is insufficient passenger flow between two unpopular stations. For the same algorithm, the performance is also different for different categories of package delivery requests. The performance for the Category I packages is the best; the performance is the worst for the Category III packages. Particularly, **maxProb** ensures that around 90% of the Category I packages (75% for Category II and III) can be arrived-on-time successfully. While for **FCFS**, only less than 10% of all Category packages can be delivered by the deadline. Similar to the the previous results, **DesCloser** performs better than **FCFS** and worse than **maxProb** for all three categories.

3) Results of Number of Relays: We compare the number of relays[7] of **maxProb** under different given deadlines, with the results shown in Fig. 9. One can observe that the number of relays presents an ascending tendency with the increase of the given deadline. One possible reason is that more *ineffective* relays (i.e., the package moves back and forth towards the destination commonly) are resulted since **maxProb** is inclined to hitchhike the coming taxi immediately, as discussed.

We also show the results on how the number of relays obtained by **maxProb** varies under different time of day in Fig. 10. Overall, the number of relays is more or less unchanged (i.e., with a value of around 3) during the whole day, except for the early morning when the number of relays is relatively smaller, probably caused by the extremely little passenger flow during that time.

We report the results of the number of relays for four algorithms w.r.t the day time and nigh time respectively, as shown in Fig. 11. As can be predicted, the number of relays for **Direct** shall equal one. For the other three algorithms,

[7]For the experiments on the number of relays, it should be noted that the number of relays is counted and averaged only for the packages that are delivered on time.

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

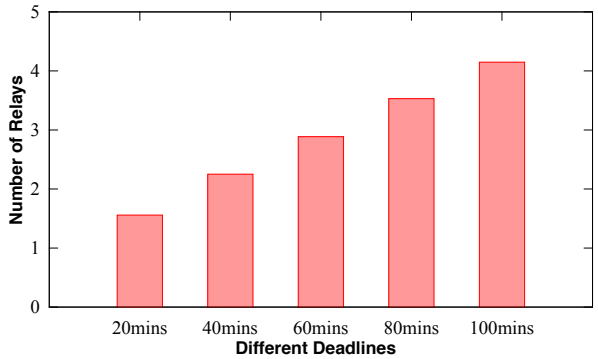


Fig. 9. Evaluation results of the number of relays under different deadlines.

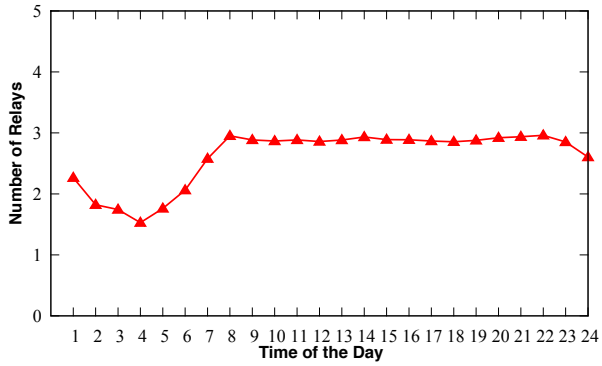


Fig. 10. Evaluation results of the number of relays under different time of the day.

slightly more relays are generally required at the day time than the night time. Moreover, a little surprisingly, the number of relays resulted from **DesCloser** and **FCFS** are quite close to that obtained by **maxProb**, implying that the number of relays is somehow independent of the adopted algorithms for a successful package delivery. We will investigate deeper about the potential causes qualitatively and quantitatively such as the geographical and temporal distributions of the successful package deliveries in the future work.

We further report the results of the number of relays for four

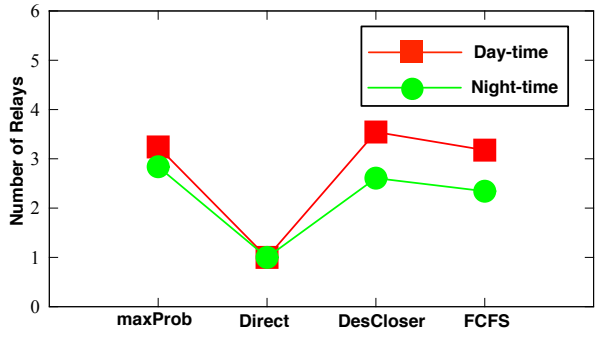


Fig. 11. Evaluation results of the number of relays for four algorithms at day time and night time respectively.

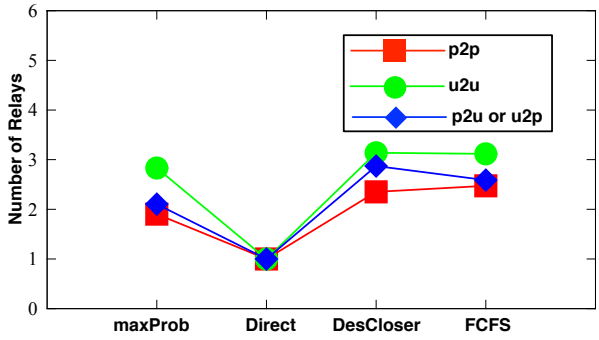


Fig. 12. Evaluation results of the number of relays for four algorithms for different package categories.

algorithms w.r.t the package categories, as shown in Fig. 12. Similarly, the number of relays for **Direct** is one, regardless of the package categories. Compared to the other two algorithms (i.e., **DesCloser** and **FCFS**), **maxProb** requires slightly fewer relays for all three package categories. Similar to the results of the success rate w.r.t the package categories, for all algorithms except for **Direct**, the performance in terms of the number of relays is the best for the Category I packages; the performance for the Category III packages is the worst.

4) Results of Package Throughput: It is necessary to evaluate the package throughput of the system, since it is a primary consideration when applied to real-life scenarios. Fig. 13 shows the results on the number of packages that the proposed system can transport successfully w.r.t the number of total generated package delivery requests per day, together with the success rate. More specifically, package throughput increases gradually with the number of generated package delivery requests before approaching a stable value. On the contrary, the success rate declines almost linearly with the number of generated package delivery requests. As can be seen, the maximum package throughput is around 20,000 per day, however, the corresponding success rate is quite low (i.e., around 40%) and might be not applicable in real life. To make the proposed system practical in real situations, the package throughput should be around 9,500 per day while maintaining a relatively promising success rate.

We further examine the package throughput under different density (number) of interchange stations, as shown in Fig. 14. As expected, the more interchange stations our system has, the higher throughput it can achieve (the number of package delivery requests for this study is 10,000). We argue that the *root cause is the improvement on utilization ratio of taxi trips for package deliveries*, which is defined as the ratio between the number of taxi trips involved in package deliveries and the total number of taxi trips. We thus plot the *utilization ratio* under different density of interchange stations in Fig. 14 as well. As evidenced, an increasing utilization ratio (35.5%, 58.3% and 64.9% respectively) is achieved as the interchange stations in the city becomes denser (18, 34 and 46 respectively). Moreover, there is still a considerable room to increase the package throughput further if placing more interchange stations to better utilize the taxi trips. We

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

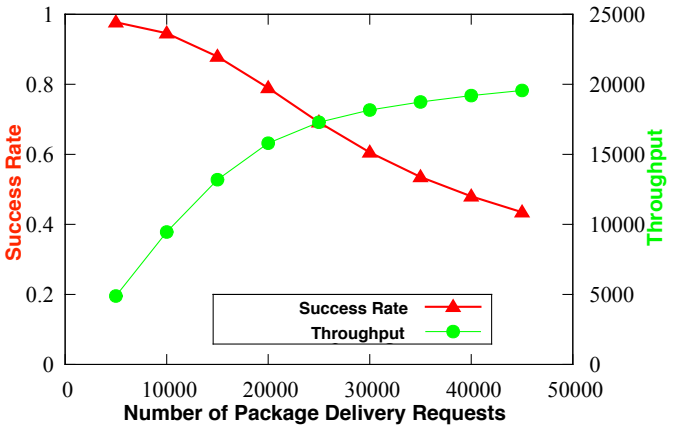


Fig. 13. Evaluation results of success rate and throughput w.r.t. package delivery requests.

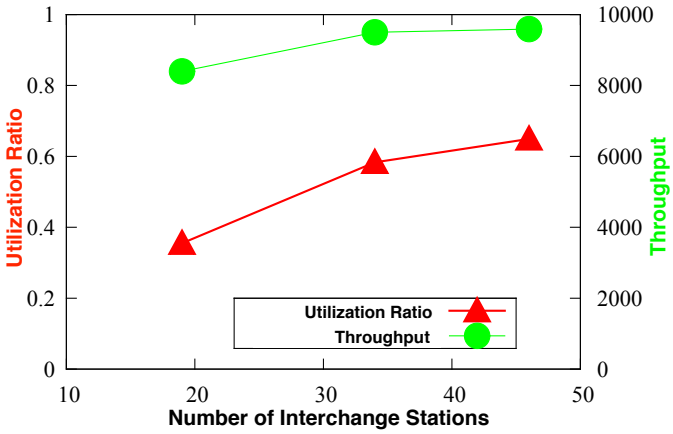


Fig. 14. Evaluation results of utilization ratio and throughput w.r.t. number of interchange stations.

are aware that the choice of interchange stations (different locations but with the same number) may affect the throughput result, but the overall trend shall be the same.

E. Open Research Issues

In this section, we discuss some open issues which are not resolved in this work but will be addressed in the future.

Package Delivery Request. Modeling package delivery request is a challenging task, requiring additional data sources, such as demographics, socio-economics, land use, and on-line purchasing activities. How to accurately model the city-wide package flow distributions can be a separate research problem itself, and there has not yet been any reliable model as far as we know [19]. In the scope of this paper, we focus on discovering the near-optimal package delivery paths with high efficiency, given any package delivery request. Hence, we simply generate the package birth time, origins and destinations randomly. In the future, we plan to incorporate other more real-world data sources to model the city-wide package delivery requests [32], and integrate them into our framework.

Transport Network Optimization. At the current research of crowd logistics, most studies focus on developing advanced package routing algorithms. In contrast, less attention has been paid to the package transport network optimization (i.e., the number/locations of interchange stations). In the future, we plan to address the two issues simultaneously.

Multi-objective Optimization. The on-time performance is one of the important optimization objectives of crowdsourced logistics. From the side of logistics service providers, other objectives are also equally important, such as the money paid to the taxi drivers. The more taxi drivers are recruited, the more rewards are necessary. Thus, minimizing the number of package relays is worth considering. From the side of taxi drivers, the minimization of detour driving distance is of great importance. In the future, we plan to formulate the problem of crowd delivery via hitchhiking taxi rides as the multi-objective optimization one.

Practical Issues. There are still many practical issues to be addressed before truly realizing the system. The maintenance cost and the potential package loss or damage at the interchange stations is one example. One promising solution to address the issue is to install the unmanned and automatic smart boxes at the stations. In this way, packages can be safely stored and drivers are required to enter one-time password to get them. The capacity issue of the interchange station is another example. On one hand, interchange stations frequently visited by passengers are more likely to be recruited for relaying packages, and thus require a larger capacity in general. On the other hand, the spatial and temporal traffic patterns of package deliveries also have a critical impact on the capacity issue [13]. Other practical issues such as the incentive mechanism for taxi drivers, the package pricing methods, the cooperation cost of the interchange station, the size of the package, the standard for container of the package also need further investigation.

VII. Conclusions and Future Work

In this paper, we present a novel framework called CrowdExpress for package delivery path planning. The framework proposed to exploit *hitchhiking rides* provided by occupied taxis to transport packages in time without degrading the quality of passenger services. More specifically, we first built the package transport network by mining the taxi GPS trajectory data offline. Then we proposed a two-phase approach for package delivery path planning with a novel and comprehensive process. Using real-world datasets which include road network data, and a large-scale taxi trajectory data generated by over 19,000 taxis in a month in NYC, US, we compared our proposed method with three baseline algorithms, and showed that our method is more efficient and effective.

In the future, we plan to broaden and deepen this work in several directions. First, we plan to correlate the package deadline to the package pricing. Currently, we set a general and relative deadline for all package deliveries and handle them equally. In the near future, we plan to set different priorities for packages according to users' expected arriving time. For instance, users are charged higher if they want their packages to be arrived earlier, and new package routing algorithms

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.

should be developed accordingly. Second, we intend to take actions to improve the system throughput and coverage. Such as grouping packages with close destinations and optimizing the interchange stations (number and location). Finally, we plan to implement and test our system with real users in actual settings, collecting feedback on how to further improve the service.

ACKNOWLEDGEMENTS

The work was supported by the National Natural Science Foundation of China (No. 61872050), the Chongqing Basic and Frontier Research Program (No. cstc2018jcyjAX0551).

REFERENCES

[1] L. Alarabi, A. Eldawy, R. Alghamdi, and M. F. Mokbel. TAREEG: a mapreduce-based web service for extracting spatial data from openstreetmap. In *Proc. of the ACM SIGMOD*, pages 897–900, 2014.

[2] T. H. Ali, S. Radhakrishnan, S. Pulat, and N. C. Gaddipati. Relay network design in freight transportation systems. *Transportation Research Part E: Logistics and Transportation Review*, 38(6):405–422, 2002.

[3] C. Archetti, M. Savelsbergh, and M. G. Speranza. The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254(2):472–480, 2016.

[4] A. M. Arslan, N. Agatz, L. Kroon, and R. Zuidwijk. Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1):222–235, 2019.

[5] R. E. Bellman and S. E. Dreyfus. *Applied dynamic programming*. Princeton university press, 2015.

[6] S. Benjaafar, Y. Li, and M. Daskin. Carbon footprint and the management of supply chains: Insights from simple models. *IEEE transactions on automation science and engineering*, 10(1):99–116, 2013.

[7] C. Chen, Z. Wang, and D. Zhang. Sending more with less: Crowdsourcing integrated transportation as a new form of citywide passenger–package delivery system. *IT Professional*, 22(1):56–62, 2020.

[8] C. Chen, D. Zhang, N. Li, and Z.-H. Zhou. B-Planner: Planning bidirectional night bus routes using large-scale taxi gps traces. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1451–1465, 2014.

[9] C. Chen, D. Zhang, X. Ma, B. Guo, L. Wang, Y. Wang, and E. Sha. CrowdDeliver: Planning city-wide package delivery paths leveraging the crowd of taxis. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1478–1496, 2017.

[10] P. Chen and S. Chankov. Crowdsourced delivery for last-mile distribution: An agent-based modelling and simulation approach. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1271–1275. IEEE, 2017.

[11] W. Chen, M. Mes, and M. Schutten. Multi-hop driver-parcel matching problem with time windows. *Flexible services and manufacturing journal*, pages 1–37, 2017.

[12] C. Cleophas, C. Cottrill, J. F. Ehmke, and K. Tierney. Collaborative urban transportation: Recent advances in theory and practice. *European Journal of Operational Research*, 273(3):801–816, 2019.

[13] T. Crainic, L. Gobbato, G. Perboli, and W. Rei. Logistics capacity planning: A stochastic bin packing formulation and a progressive hedging metaheuristic. Technical report, CIRRELT-2014-66, 2014.

[14] A. Devari. Crowdsourced last mile delivery using social networks. Master's thesis, The State University of New York at Buffalo, 2016.

[15] J. Du, B. Guo, Y. Liu, L. Wang, Q. Han, C. Chen, and Z. Yu. CrowDNet: Enabling a crowdsourced object delivery network based on modern portfolio theory. *IEEE Internet of Things Journal*, 6(5):9030–9041, 2019.

[16] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of ACM KDD*, volume 96, pages 226–231, 1996.

[17] E. Fatnassi, J. Chaouachi, and W. Klibi. Planning and operating a shared goods and passengers on-demand rapid transit system for sustainable city-logistics. *Transportation Research Part B: Methodological*, 81:440–460, 2015.

[18] B. L. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval research logistics*, 34(3):307–318, 1987.

[19] J. González-Feliu, M. G. Cedillo-Campo, and J. L. García-Alcaraz. An emission model as an alternative to od matrix in urban goods transport modelling. *Dyna*, 81(187):249–256, 2014.

[20] N. Kafle, B. Zou, and J. Lin. Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery. *Transportation Research Part B: Methodological*, 99:62–82, 2017.

[21] B. Li, D. Krushinsky, H. A. Reijers, and T. Van Woensel. The share-a-ride problem: People and parcels sharing taxis. *European Journal of Operational Research*, 238(1):31–40, 2014.

[22] B. Li, D. Krushinsky, T. Van Woensel, and H. A. Reijers. The share-a-ride problem with stochastic travel times and stochastic delivery locations. *Transportation Research Part C: Emerging Technologies*, 67:95–108, 2016.

[23] Y. Liu, B. Guo, C. Chen, H. Du, Z. Yu, D. Zhang, and H. Ma. FooDNet: Toward an optimized food delivery network based on spatial crowdsourcing. *IEEE Transactions on Mobile Computing*, 18(6):1288–1301, 2018.

[24] R. Lowe and M. Rigby. The last mile–exploring the online purchasing and delivery journey. Technical report, Barclays, 2014.

[25] J. McInerney, A. Rogers, and N. R. Jennings. Crowdsourcing physical package delivery using the existing routine mobility of a local population. In *the Orange D4D Challenge*, 2014.

[26] Y. Nie and Y. Fan. Arriving-on-time problem: discrete algorithm that ensures convergence. *Transportation Research Record: Journal of the Transportation Research Board*, (1964):193–200, 2006.

[27] A. Punel, A. Ermagun, and A. Stathopoulos. Studying determinants of crowd-shipping use. *Travel Behaviour and Society*, 12:30–40, 2018.

[28] A. J. Rohm and V. Swaminathan. A typology of online shoppers based on shopping motivations. *Journal of Business Research*, 57(7):748 – 757, 2004.

[29] A. Sadilek, H. Kautz, and J. P. Bigham. Finding your friends and following them to where you are. In *Proc. of WSDM*, pages 723–732, 2012.

[30] A. Sadilek, J. Krumm, and E. Horvitz. Crowdphysics: Planned and opportunistic crowdsourcing for physical tasks. In *Proc. ICWSM*, 2013.

[31] S. Skiena. Dijkstra's algorithm. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley*, pages 225–227, 1990.

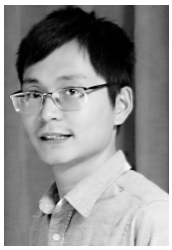
[32] X. Tan, Y. Shu, X. Lu, P. Cheng, and J. Chen. Characterizing and modeling package dynamics in express shipping service network. In *Proc of IEEE Congress on Big Data*, pages 144–151, 2014.

[33] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

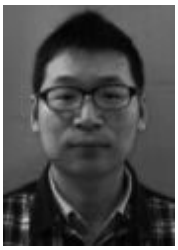
[34] H. Üster and P. Kewcharoenwong. Strategic design and analysis of a relay network in truckload transportation. *Transportation Science*, 45(4):505–523, 2011.

[35] X. Zheng, X. Liang, and K. Xu. Where to wait for a taxi? In *Proc. of ACM UrbComp*, pages 149–156, 2012.

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.



Chao Chen is a full professor at College of Computer Science, Chongqing University, Chongqing, China. He obtained his Ph.D. degree from Pierre and Marie Curie University and Institut Mines-Télécom/Télécom SudParis, France in 2014. He received the B.Sc. and M.Sc. degrees in control science and control engineering from Northwestern Polytechnical University, Xi'an, China, in 2007 and 2010, respectively. Dr. Chen got published over 80 papers including 20 ACM/IEEE Transactions. His work on taxi trajectory data mining was featured by IEEE Spectrum in 2011 and 2016 respectively. He was also the recipient of the Best Paper Runner-Up Award at MobiQuitous 2011. His research interests include pervasive computing, mobile computing, urban logistics, data mining from large-scale GPS trajectory data, and big data analytics for smart cities.



Sen Yang was a master student at College of Computer Science, Chongqing University, Chongqing, China. His research interests include scenic travel route planning, taxi GPS trajectory data mining, and last-mile delivery.

Yasha Wang received his Ph.D. degree in Northeastern University, Shenyang, China, in 2003. He is a professor and associate director of National Research & Engineering Center of Software Engineering in Peking University, China. His research interest includes urban data analytics, ubiquitous computing, software reuse, and online software development environment. He has published more than 50 papers in prestigious conferences and journals, such as ICWS, UbiComp, ICSP and etc. As a technical leader and manager, he has accomplished several key national projects on software engineering and smart cities. Cooperating with major smart-city solution providing companies, his research work has been adopted in more than 20 cities in China.

Bin Guo is a professor from Northwestern Polytechnical University, China. He received his Ph.D. degree in computer science from Keio University, Japan in 2009 and then was a post-doc researcher at Institut Mines-TELECOM/TELECOM SudParis in France. His research interests include ubiquitous computing, mobile crowd sensing, and HCI.

Daqing Zhang is a professor at Institute of Software, School of Electronics Engineering and Computer Science, Peking University, China. He obtained his Ph.D from University of Rome "La Sapienza" and the University of L'Aquila, Italy in 1996. His research interests include large-scale data mining, urban computing, context-aware computing, and ambient assistive living. He has published more than 180 referred journal and conference papers, all his research has been motivated by practical applications in digital cities, mobile social networks and elderly care.

Dr. Zhang is the Associate Editor for four journals including *ACM Transactions on Intelligent Systems and Technology*. He has been a frequent Invited Speaker in various international events on ubiquitous computing. He is the winner of the Ten Years CoMoRea Impact Paper Award at IEEE PerCom 2013, the Best Paper Award at IEEE UIC 2015/2012 and the Best Paper Runner Up Award at Mobiquitous 2011. He is a member of the China Thousand-Talent Program.

Authorized licensed use limited to: Telecom SudParis (Frmly Telecom et management SudParis INT). Downloaded on November 12,2020 at 16:57:19 UTC from IEEE Xplore. Restrictions apply.