

coRide: Carpool Service with a Win-Win Fare Model for Large-Scale Taxicab Networks ^{*}

Desheng Zhang[†]
zhang@cs.umn.edu

Ye Li[‡]
li.ye@siat.ac.cn

Fan Zhang[‡]
zhangfan@siat.ac.cn

Mingming Lu[‡]
lumm@siat.ac.cn

Yunhuai Liu[◊]
yunhuai@trimps.ac.cn

Tian He[†]
tianhe@cs.umn.edu

[†]Department of Computer Science and Engineering, University of Minnesota, USA

[‡]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China

[◊]Third Research Institute of Ministry of Public Security, China

Abstract

Carpooling has long held the promise of reducing gas consumption by decreasing mileage to deliver co-riders. Although *ad hoc* carpools already exist in the real world through private arrangements, little research on the topic has been done. In this paper, we present the first systematic work to design, implement, and evaluate a carpool service, called *coRide*, in a large-scale taxicab network intended to reduce total mileage for less gas consumption. Our *coRide* system consists of three components, a dispatching cloud server, passenger clients, and an onboard customized device, called *TaxiBox*. In the *coRide* design, in response to the delivery requests of passengers, dispatching cloud servers calculate cost-efficient carpool routes for taxicab drivers and thus lower fares for the individual passengers.

To improve *coRide*'s efficiency in mileage reduction, we formulate a NP-hard route calculation problem under different practical constraints. We then provide (i) an optimal algorithm using Linear Programming, (ii) a 2 approximation algorithm with a polynomial complexity, and (iii) its corresponding online version. To encourage *coRide*'s adoption, we present a win-win fare model as the incentive mechanism for passengers and drivers to participate. We evaluate *coRide* with a real world dataset of more than 14,000 taxicabs, and the results show that compared with the ground truth, our service can reduce 33% of total mileage; with our win-win fare model, we can lower passenger fares by 49% and simultaneously increase driver profit by 76%.

Categories and Subject Descriptors

H.4 [Information System Application]: Miscellaneous

General Terms

Algorithms, Design, Experimentation, Performance

^{*}Prof. Tian He is the corresponding author of this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SenSys '13, November 11 - 15, 2013, Roma, Italy
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2027-6/13/11 ...\$15.00
<http://dx.doi.org/10.1145/2517351.2517361>

1 Introduction

Among all transportation modes, taxicabs play a particularly prominent role in residents' daily commutes in many metropolitan areas [2] [6]. Based on a recent survey in New York City [15], over 100 taxicab companies operate more than 13,000 taxicabs, with stable demand of 660,000 passengers per day, and transport more than 25% of all transit passengers, accounting for 45% of all transit fares paid. To fulfill such delivery requests, these taxicabs travel a total of roughly 800 million miles per year [6]. Unfortunately, with 25 MPG, these taxicabs consume about 32 million gallons of gas every year, more than the total annual gas consumption in some middle-sized countries (*e.g.*, Central African Republic [13]), therefore leading to severely harmful tailpipe emissions and energy consumption. On the other hand, in carbon emissions trading under the Kyoto Protocol [18], governments will provide economic incentives for achieving reductions in the emissions of carbon pollutants. Thus, for both environmental and economic purposes, it is imperative to find a practical initiative to support the same delivery requests for taxicab transportation from passengers with lower total mileage and less carbon emissions.

In this paper, we argue that a *taxicab carpool service* is a promising solution. The key advantage of a carpool service is that it can pool groups of several passengers heading in the similar direction into *one* rather than *several* taxicabs. In other words, a carpool service provides a valid solution for delivering the same number of passengers with lower total mileage and thus less gas. The economic incentive for drivers is that groups of passengers can pay a higher aggregated fare, whereas the incentive for passengers is that every passenger will pay less than in a non-carpool situation. With an effective fare model, we can achieve a win-win situation. Furthermore, carpools can also improve the availability of taxicab service during rush hours and after major events.

Admittedly, taxicab carpooling is not a new concept and has been around for years. But existing taxicab carpool arrangements are negotiated by individual drivers and passengers in an *ad hoc* manner without a facilitating infrastructure. Until now, we have lacked a systemic study of carpooling in large scale taxicab networks. We note that many studies have focused on taxicab scheduling [4] [9] [10] [11] [19] [23] [24] or novel systems taking advantage of taxicab mobile traces [3] [5] [12] [17] [20] [21]

[22] [25] [26], but little research, if any, has been done on taxicab carpool services with a software and hardware co-design. In this paper, we present the first systematic study of how to design, evaluate, and implement taxicab carpool services in real world scenarios. Specifically, the key contributions of this paper are as follows:

- To the best of our knowledge, we conduct the first carpool service that considers the mutual benefits for passengers and drivers in large-scale taxicab networks and provide a comprehensive study of how to fill the same passenger delivery requests with less total mileage and thus less gas consumption. To achieve our goal, we develop customized hardware, TaxiBox, with multiple sensors and onboard devices (such as CDMA communication module, MIC, camera, and carpool fare meters). Using TaxiBox, we design a taxicab carpool system, *coRide*, to gather requests from passengers, to inform drivers of carpool requests, to calculate carpool routes for drivers, and to estimate carpool fares for passengers.
- In *coRide*, we introduce a mathematical concept *delivery graph* to represent a carpool route schedule for delivering passengers. Given requests provided by passengers, we seek an optimal delivery graph to achieve the minimum total mileage. We show that this optimization is NP-hard by linking it to the classic traveling salesman problem, and provide (i) an optimal solution with integer programming, (ii) a 2-factor approximation solution with a polynomial complexity, and (iii) an online algorithm to accommodate online streaming requests. In addition, we consider different real world constraints, *e.g.*, passenger travel periods, number of available taxicabs, and taxicab capacities.
- Based on the carpool route, we propose a win-win carpool fare model to encourage both drivers and passengers to participate in carpooling. In this model, given a carpool benefit due to mileage reduction, passengers and a driver will share this benefit based on a ratio dynamically adjusted by the supply and demand relationships in a taxicab network.

Our evaluation effort is comprehensive. We test *coRide* on a real world 10 GB dataset consisting of 1 week of GPS traces from more than 14,000 taxicabs in a Chinese city Shenzhen with a population of 10 million. The evaluation results show that compared with the ground truth, our carpool service can reduce the total mileage by as much as 33%, and our win-win fare model can lower passenger fares by 49% and increase driver profit by 76% at the same time.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 proposes the motivation for the design. Section 4 presents the motivation for the design. Section 4 presents the *coRide* system overview. Section 5 describes our customized device, TaxiBox. Section 6 explains the algorithms for carpool route calculation. Section 7 presents a win-win carpool fare model. Section 8 validates our services with a big dataset, followed by the conclusion in Section 9.

2 Related Work

The premise of taxicab carpooling is not new, but in real world it is normally negotiated privately by drivers and passengers in an *ad hoc* manner. We lack a systematic design to balance benefits of all the involved parties, *e.g.*, drivers, passengers, and taxicab operators. Two types of previous work are directly related to our work: research on taxicab systems and on *ad hoc* carpools.

2.1 Taxicab Systems

The increasing availability of GPS devices has encouraged a surge of research intended to improve the efficiency of large-scale taxicab networks. First, several systems are proposed for the benefit of passengers or drivers, *e.g.*, allowing passengers to query the expected duration and fare of a planned taxicab trip based on the history of previous trips [4] and query real-time taxicab availability to make informed transportation choices [19], as well as recommending optimal pickup locations or routes [8] [9] [23]. Second, taxicab traces can also help taxicab network operators better oversee taxicabs and provide efficient service to passengers, *e.g.*, discovering spatial and temporal causal interactions to provide timely and efficient service in certain areas with disequilibrium [10] [11], and detecting anomalous taxicab trips to discover driver fraud or road network changes [24]. Third, traces from experienced taxicab drivers can help other drivers improve their driving performance, *e.g.*, navigating newer drivers to smart routes based on those of experienced taxicab drivers [17] [21]. Fourth, large scale taxicab traces enable us to better understand traffic conditions of cities, *e.g.*, semantics of origin-destination flows [25], traffic congestion and volumes [3], and traffic patterns between regions with different functions [20]. Finally, large scale traces also can help in city planning, *e.g.*, detecting flawed urban planning [26] or improving map inference [5] [12].

Yet existing research on taxicab systems focuses on scheduling individual taxicabs, assuming that one taxicab can accommodate only a single delivery request at a time. In contrast, our system allows shared delivery. Technically, we focus on carpool route calculation and a win-win fare model, neither of which has been investigated before.

2.2 Ad Hoc Carpool

Limited *ad hoc* taxicab carpools exist in both developed and developing countries. For example, in New York City, up to four passengers can carpool together in a single taxicab ride during 6 AM to 10 AM on a weekday, along three preset routes in Manhattan at a flat fare of \$3 or \$4 per passenger, significantly less than the regular metered rates [16]. In Beijing, *ad hoc* taxicab carpooling is also allowed with the consent of both passengers and drivers, and every passenger pays 60% of the regular fare. Further, some door to door shuttle services are also available in major airports, and can enable shared rides to or from airports [1]. However, in the aforementioned carpool services, both time and locations are preset and the services are arranged on the spot by passengers or drivers in a small-scale *ad hoc* manner, and no infrastructure is provided to improve the efficiency of carpooling.

3 Motivation

In this section, based on two datasets about traces and fares of 14,000 taxicabs in a Chinese city Shenzhen with a population of 10 million, we first introduce the basic properties of large scale taxicab networks in dense urban areas. Then, we present evidence about the inefficiencies of current taxicab networks, demonstrate opportunities for car-pools to address these inefficiencies, and identify challenges to facilitating carpooling in current taxicab networks. Details of the datasets appear in Section 8.

3.1 Properties of Taxicab Networks

In developed countries such as US, taxicabs are usually used to serve passengers to airports, and personal vehicles are used for other activities, excepting extreme large cities such as New York City. But in developing countries, due to high costs of owning personal vehicles, taxicabs and other public transportation are popular for daily activities. In dense urban areas such as Beijing, taxicabs are affordable for local traveling with an initiate fare about 2 USD for a 3 KM trip, and are more comfortable than other public transportation with cheaper fares (such as buses or subway). Due to the popularity and the affordability of taxicab services, the number of taxicabs in a taxicab network of a large city is typically more than 10,000. Thus, these taxicabs can be easily found on streets at the most of time and locations (except in rush hours or in hot pickup spots) and are commonly used for shopping, traveling to and from the work or schools, and other daily activities.

3.2 Inefficiencies of Taxicab Networks

For society, the key inefficiency of taxicab networks is the large gas consumption of a *long-travel distance*. Figure 1 summarizes statistics about taxicab network studied. We observe that these taxicabs travel a total of 1.2 billion kilometers per year, consuming about 100 million liters of gas to deliver 200 million passengers and causing harmful tailpipe emissions.

Taxicab Network Summary	
Collection Period	6 Months
Collection Date	01/01/12-06/30/12
Number of Taxicabs	14,453
Number of Passengers	98,472,628
Total Travel Distance	594,031,428 (KM)
Total Fare	2,255,052,932 (CNY)
Average Travel Distance	6.032 (KM)
Average Fare	22.9 (CNY)

Fig 1. Statistics

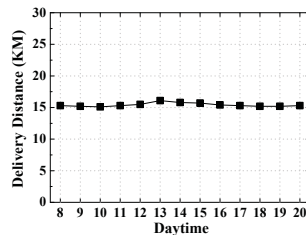


Fig 2. Delivery Distance

For drivers, the key inefficiency is *low profits*, which are decided mainly by delivery distances (*i.e.*, the mileage with paying passengers). Intuitively, drivers should earn more profit in rush hours, but this is not the case in large cities with severe congestion. In regular hours without congestion, the total distance (*i.e.*, also including total mileage without paying passengers) is high, but the percentage of delivery distance is low, since it is not easy to find a passenger. In rush hours with congestion, in contrast, the percentage of delivery distance is high (*i.e.*, easy to find a passenger), but the total distance is low due to the slow pace of traffic. Figures 2 shows the average delivery distance. It

shows delivery distances at different times of day (*i.e.*, rush and non-rush hours) are not significantly different.

For passengers, key inefficiencies are *high fares* and *low availability*. According to statistics about New York City [6] [15], the average fare is 11.44 USD for a 2.8 mile trip with an 11-minutes travel time, which is 5.8 times higher than the average public transit fare (bus or subway) on average. In our statistics, the average fare of 22.9 Chinese Yuan (CNY) for taxicabs is 11 times of a bus fare of 2 CNY on average. These two datasets also provide some evidence about the low availability of taxicab services. First, in the time intervals between deliveries presented in by Figure 3, a small interval indicates that a taxicab will pick up a new passenger right after it drops off an old passenger, *i.e.*, low availability. In Figure 3, the average time interval in rush hours is small, less than 3 minutes, indicating a low availability of taxicab services. Second, low availability can also be seen in the taxicab occupancy ratios in Figure 4, where a high occupancy ratio indicates fewer empty taxicabs. The ratios in Figure 4 indicate that more than 80% of taxicabs are occupied on average during rush hours. Thus, Figures 3 and 4 indicate the low availability of taxicabs during rush hours.

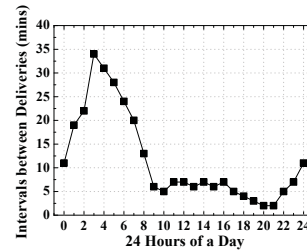


Fig 3. Delivery Intervals

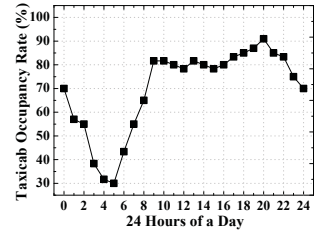


Fig 4. Occupancy Rate

3.3 Opportunities for Taxicab Carpool

We show the opportunities that carpools provide to address the above inefficiencies. A key factor ensuring the success of carpooling is the frequency with which passengers from the same origin go to similar destinations, which can be shown by (i) travel distances of shared routes and (ii) distances between destinations. Both long-distance shared routes and small distances between passengers' destinations indicate good carpool opportunities. We first show some examples of shared routes and close destinations. Based on the datasets, we show 200 continuous deliveries from an airport in Figure 5 where in ordinary services, most passengers go downtown and others go to several hot spots.

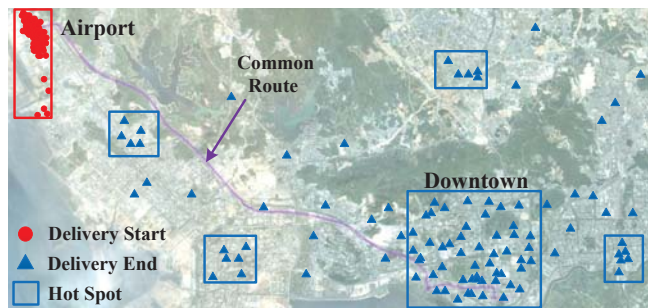


Fig 5. Deliveries From Airport

Further, with 1,000 taxicab deliveries from the airport, Figure 6 shows the percentage distribution of distances of shared routes, where we observe that more than 90% of deliveries share at least 5 KM with another delivery, and that more than 50% of deliveries share at least 20 KM with another delivery. The percentage distribution of distances between destinations is given in Figure 7, where we observe that almost 60% of deliveries have a destination closer than 1 KM to that of another delivery, and almost 80% of deliveries have a destination closer than 5 KM to another destination. From Figure 6 and 7, we observe that a fair amount of deliveries share long distances, and have destinations close to each other.

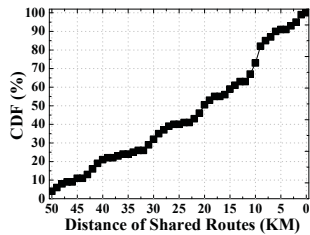


Fig 6. Shared Distance

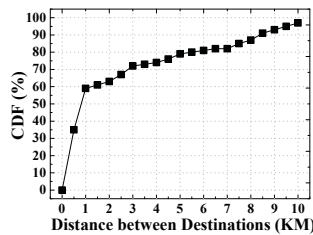


Fig 7. Close Destinations

3.4 Benefits for Taxicab Carpool

From the above figures, we observe a good opportunity for carpools to benefit multiple parties. For passengers, a taxicabs carpool can increase the availability of taxicab services in extreme weathers, peak hours or hot pickup locations, reducing the waiting time for passengers; in addition, multiple passengers in a carpool can share the fare together, reducing the fare paid by individual passengers. For taxicab drivers, a taxicab carpool can increase profits, since the aggregated carpool fare is higher than regular service fare with the same travel distance. For operators, a taxicab carpool can provide more transportation capacity and enable more efficient gas consumption. Note that taxicab carpools do not aim to completely replace the traditional taxicab services, but serve as a key supplement for the situations where regular taxicab services are insufficient in peak hours or extreme weathers, or situations where some passengers would like to take transportation that is cheaper than traditional taxicab services yet more convenient than bus and subway.

3.5 Passenger Sensitivity for Taxicab Carpool

According to a carpool survey taking at Beijing [7], 75% of interviewees are willing to carpool; 57% of interviewees have carpooled at least once; 73% of interviewees accept a simple carpool fare model that every passenger pays 60% of regular service fare for the shared distance, leading to more profits for drivers; several key concerns about carpooling pointed out by more than half of interviewees are as follows (i) prolonged travel time (64%), (ii) hard to find passengers to carpool (50%), and (iii) unable to print duplicated receipts for all passengers (50%). Based on the above survey, we find that most passengers are willing to accept carpools and to share the benefits of carpools with co-riders and the driver, but we still face several challenges to enable a practical carpool service in large-scale taxicab networks, which we will introduce next.

3.6 Challenges for Taxicab Carpool

We present three challenges and some possible solutions for implementing carpool services in the current taxicab networks.

Acquisition of Detailed Status about Taxicabs: To find the most suitable taxicab, a dispatching center should acquire detailed information about taxicabs, *e.g.*, how many seats are left, which is difficult to obtain in current taxicab networks, where only the general taxicab status (*e.g.*, locations, speeds, with passenger or not, etc) can be obtained by dispatching centers through real-time GPS record uploading. Thus, an onboard device should be installed in taxicabs to let dispatching centers monitor the detailed status of taxicabs and find the most suitable taxicab.

Carpool Route Calculation: After finding the most suitable taxicab, a dispatching center should calculate the optimal carpool route based on multiple received requests in a centralized way, and send the calculated route to a driver. This route should be efficient in terms of the total distance to deliver all assigned passengers. In addition, the calculation of routes should be fast enough to enable a responsive taxicab carpool service.

Fare Estimation and Calculation: With a carpool route schedule, dispatching centers should notify passengers with fare details of several carpool options for their approval. A win-win carpool fare model that estimates fares is missing in the taxicab business. Further, current fare meters can calculate only a single fare, and a more advanced fare meter that can record multiple concurrent trips is desirable for carpooling.

To address the above challenges, we aim to develop a carpool system, *coRide*, as a hardware and software co-design with an front-end onboard device, *TaxiBox*, and a back-end cloud server to upgrade current taxicab networks. *coRide* employs multiple sensors and external devices attached to *TaxiBox* to effectively manage taxicab networks. We will provide an overview of *coRide* and the *TaxiBox* design in Sections 4 and 5, respectively.

4 The *coRide* System Overview

In this section, we present a system overview of *coRide*, which consists of three key parts: a cloud server, passengers client, and the onboard *TaxiBox* as shown in Figure 8.

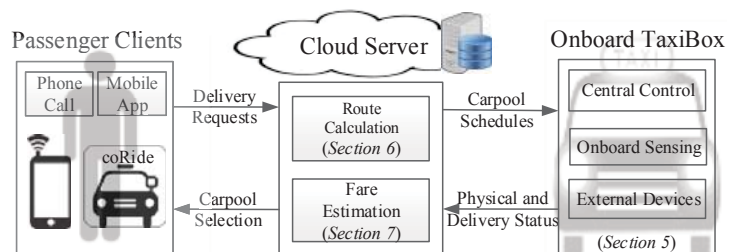


Fig 8. *coRide* System Overview

4.1 Passenger Clients

Passenger participation is required by our design, which can be encouraged by our win-win fare model discussed later. Assuming that passengers will be willing to participate, they will provide delivery requests to the dispatching center.

The most common way to provide delivery requests is to call the dispatching center by phone to provide the number of passengers, pickup time, origin, destination, and possible delivery deadline. Further, mobile apps can be used to provide delivery requests without calling the center. Based on the delivery requests provided by passengers, the dispatching center will return a carpool option with a reduced fare for their approval, along with a non-carpool option with a regular fare for comparison.

4.2 Onboard TaxiBox

When a carpool is approved by passengers, a dispatching center will locate a suitable taxicab for the carpool based on the current status of taxicabs and then send a carpool route schedule to this taxicab's TaxiBox. The driver will respond to this carpool request by changing the status of the taxicab and then performing the carpool route schedule. These functions are performed by three key components of TaxiBox, which will be introduced in detail in Section 5.

4.3 Dispatching Cloud Server

In this paper, we will focus on function designs for a cloud server at dispatching centers with an emphasis on taxicab carpool services rather than regular services. In our carpool service design, a cloud server is mainly in charge of

- (i) receiving delivery requests from passengers;
- (ii) calculating carpool routes based on delivery requests;
- (iii) estimating carpool fares for passengers to approve;
- (iv) sending carpool routes to suitable taxicabs;
- (v) obtaining the physical and delivery status of taxicabs.

In the rest part of the paper, we present the detailed TaxiBox design in Section 5; for the dispatching cloud server, we describe two key functions, the carpool route calculation and fare model in Sections 6 and 7, respectively.

5 TaxiBox Hardware

In this section, we present our hardware design, and then show the deployment about TaxiBox, and final propose the capability of taxicabs with TaxiBox.

5.1 TaxiBox Design

As shown by Figure 9, our onboard device, TaxiBox, consists of three main parts: central control system, onboard sensing system, and external devices.

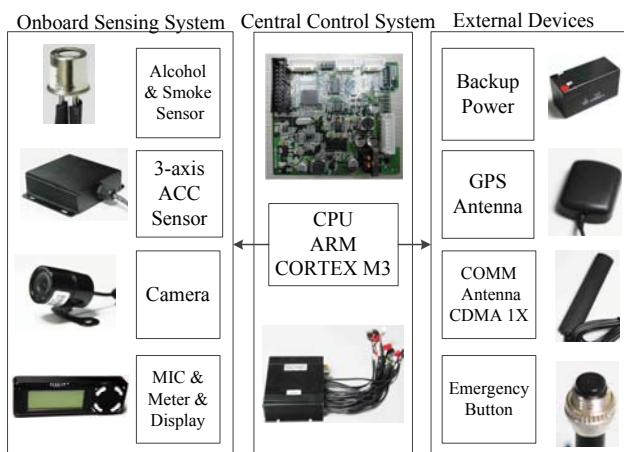


Fig 9. TaxiBox Hardware Design

The central control system has two key parts, the power module and the CPU module. For the power module, we employ TPS54160 from Texas Instruments, which is a 60V, 1.5A, step down SWIFT DC/DC converter with an integrated high-side MOSFET. For the CPU module, we use a 32 bit 72 MHz processor STM32F103 from ARM Cortex-M3 processors with A/D Convertors of 12-bit accuracy.

The onboard sensing system has open interfaces to multiple sensors, and the current hardware is attached with (i) alcohol and smoke sensors, (ii) a $\pm 2g$ triaxial acceleration sensor, and (iii) a camera and a microphone. Based on the above onboard sensors, a dispatching center is capable of monitoring the comprehensive physical status of a taxicab on streets.

Various external devices can be integrated into our TaxiBox. Some external devices in the current TaxiBox design include (i) a display and a speaker integrated to the display; (ii) a traditional fare meter for fare calculation and receipt printing; (iii) backup power for a situation in which the main power is not available; (iv) an emergency button; (v) a GPS module with a separate GPS antenna; and (vi) a CDMA 1X communication module with a separate antenna.

In some existing taxicab networks, the communication modules usually use GPRS (*e.g.*, for GPS coordinates uploading) between taxicabs and dispatching centers. But in our design, taxicabs typically have a larger dataset to upload to or download from a dispatching center. Thus, a CDMA 1X, instead of GPRS, communication module is attached to TaxiBox, since CDMA employs different channels for voice and data communications, which clearly has advantages in terms of communication speed and stability, compared to GPRS that employs the same channel for data communications.

5.2 TaxiBox Deployment

We have deployed our TaxiBox in 98 taxicabs as shown by Figure 10. The alcohol and smoke sensors are installed in the ceiling of taxicabs for better sensor function. The camera is in front of passengers so as to take pictures from a better angle. The main part of TaxiBox is hidden above the glove box. The display is installed above the air-conditioner control panel for easier access by drivers. The 3 axis acceleration sensor is hidden under the glove box.



Fig 10. TaxiBox Deployment

5.3 TaxiBox Capability

In this section, based on the hardware we deployed in taxicabs, we introduce the capabilities of TaxiBox.

Taxicab Physical Status Sensing: Dispatching centers should be fully aware of the status (*e.g.*, location, speed, *etc.*) of taxicabs to provide better carpool service. With GPS and CDMA 1X modules onboard, a taxicab can periodically upload its real-time physical status to a dispatching center. The onboard traditional fare meter and TaxiBox with a display can function together as a smart meter that can record the status of several trips, *i.e.*, the delivery distance and fare for different passengers onboard, whereas the traditional fare meter can only record a single delivery distance. Further, a speaker is integrated into the display so dispatching centers can issue a voice schedule or voice navigation.

Taxicab Delivery Status Sensing: In addition to a taxi's physical status, a dispatching center is also interested in the real time status of its deliveries. The status of deliveries includes delivery distance, with passengers or not, fare, duration, start time, end time, pickup and dropoff location, which all can be obtained by TaxiBox and uploaded to dispatching centers.

6 Route Calculation in Cloud Server

In this section, we first propose preliminaries about our carpool work, then define a carpool route calculation problem, and finally propose its solution.

Carpools can be classified into four categories, (i) one origin to one destination ($1 \rightarrow 1$); (ii) one origin to many destinations ($1 \rightarrow N$); (iii) many origins to one destination ($N \rightarrow 1$); and (iv) many origins to many destinations ($N \rightarrow N$).

For the sake of presentation, we will focus on $1 \rightarrow N$ because (i) $1 \rightarrow 1$ is a special case of $1 \rightarrow N$; (ii) $N \rightarrow 1$ can be solved with $1 \rightarrow N$ by reversing origin and destination; and (iii) $N \rightarrow N$ can be solved with a special $1 \rightarrow N$ with constraints on the order in which to visit all origins and destinations. Without loss of generality, we use $1 \rightarrow N$ model (*e.g.*, carpool passengers from an airport) as an example for the design.

6.1 Preliminaries

For a carpool, a passenger will provide a *delivery request* with an origin, a destination, a start time and an optional end time (A possible end time serves as a deadline for delivery, but our model works with an unknown end time). Thus, given several requests for carpooling from the same origin as in Figure 11(a), we shall analyze distances between their destinations, which can be shown as a complete graph. We construct this complete graph as shown in Figure 11(b) by (i) treating both origin and destinations as vertices, and (ii) linking all vertices to each other with directed edges, associated edge weights with pairwise mileage costs.

Subfigures (a) and (b) in Figure 11 give an example of how to create a complete graph based on 9 delivery requests from the origin a . A weight on an edge (*e.g.*, M_{ij}) indicates the real world mileage between two locations. Given the complete graph, we can obtain a carpool route based on a *delivery graph*, which is defined as follows.

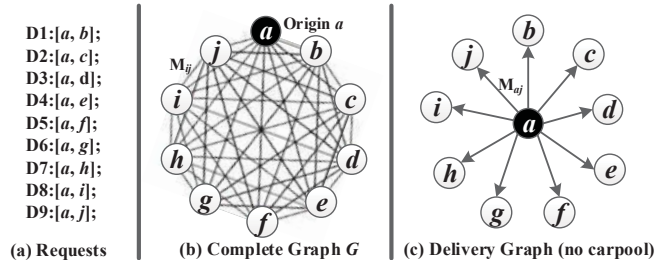


Fig 11. Complete Graph

Definition 1: Delivery Graph: With a complete graph G given by delivery requests, a *delivery graph* is a subgraph of G where (i) the origin vertex can reach all destination vertices; and (ii) no branches exist at any vertex other than the origin vertex (*i.e.*, spoke topology)

With the above definition, we can see that a delivery graph uniquely indicates a carpool route where the total carpool mileage is equal to the sum of all its edges' weights. In Definition 1, the condition (i) is to make sure that with a carpool route, all passengers can be delivered from the origin to their destinations; the condition (ii) is to make sure that every passenger will take only one taxicab during the carpool, *i.e.*, without relay. Subfigure (c) in Figure 11 gives an example of a delivery graph without carpool, *i.e.*, all passengers are delivered by separate drivers with separate mileages, *e.g.*, M_{aj} .

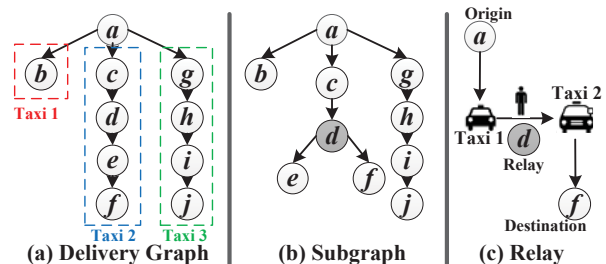


Fig 12. Delivery Graph with Carpool

The examples of a delivery graph DG with carpool are given in Subfigure (a) of Figure 12. In DG , the origin vertex a can reach all destination vertices, and no branches exist at any destination vertex. A delivery graph (*e.g.*, DG) indicates a real world carpool by specifying (i) a *passenger assignments* for taxicabs, and (ii) a *delivery order* for a taxicab's passenger assignment. For example, DG shows that three taxicabs fulfill passenger requests with destinations on three paths (the total weight on edges of a path indicates the real world mileage): Taxi 1 delivers passengers to b , with a mileage M_{ab} ; Taxi 2 delivers passengers to c, d, e and f , with a mileage $M_{ac} + M_{cd} + M_{de} + M_{ef}$; Taxi 3 delivers passengers to g, h, i and j , with a mileage $M_{ag} + M_{gh} + M_{hi} + M_{ij}$.

Subfigure (b) of Figure 12 gives an example of carpools prohibited by *coRide*. To carpool by this subgraph, we have to at least use two taxicabs to deliver passengers to c, d, e and f : the first taxicab delivers passengers with destination

c , d , and e , but carries only the passenger with destination f from origin a to an intermediate vertex d ; the second taxicab has to pick up this passenger at vertex d (a relay), and then deliver him or her to destination f as shown in Subfigure (c).

We argue that the carpool service with a relay is not practical in real world scenarios, because (i) the relayed passenger has to pay multiple times to different drivers, and (ii) the coordination between taxicabs would lead to a large layover delay. Therefore, in our work, *coRide* supports only the delivery graphs with spoke topology to indicate a practical real world carpool route schedule for drivers to deliver passengers without a relay.

6.2 Carpool Route Calculation Problem

Based on the delivery graph proposed in the last subsection, we propose our carpool route calculation problem: **Given a complete graph based on delivery requests, find the minimum weight delivery graph.**

The complete graph can be easily constructed based on delivery requests provided by passengers; as a subgraph, a delivery graph specifies passenger assignments and delivery orders to fulfill all delivery requests; the minimum total weight of a delivery graph indicates it fulfills all requests with a carpool spending the minimum total mileage. To perform a practical carpool, we also consider three constraints for our design.

(i) **Taxicab Capacity** c ; it shows how many passengers can be pooled into one taxicab.

(ii) **Number of Available Taxicabs** n ; it shows how many taxicabs can be used for carpool at the origin.

(iii) **Travel Period** $[t_i^s, t_i^e]$; it shows the earliest pickup time t_i^s and the latest dropoff time t_i^e for a delivery request i .

Based on the above discussion, our carpool route calculation problem is related to a *multiple traveling salesmen problem* (called mTSP where multiple salesmen start from a depot to visit different cities with the minimum total distance [14]) yet with the special carpool constraints. An mTSP is generally solved with Integer Linear Programming to the optimal solution. But for our large scale carpool route calculations, the optimal solution results in a long running time, since it is NP-Hard. Thus, a practical approximation algorithm should be used to obtain a delivery graph within a reasonable time.

Another key feature of our carpool route calculation problem is that instead of booking a carpool trip a day or two in advance, some passengers may provide *online* delivery requests just tens of minutes before the starting time of their deliveries. So an *online* algorithm should be employed to pool new passengers into existing carpools or to start a new carpool.

Therefore, the design agenda about our solution to the carpool route calculation problem given as follows:

(i) we use Integer Linear Programming to formulate our carpool route calculation to obtain the optimal solution in Section 6.3;

(ii) for a practical (quicker) solution, we propose a 2-factor approximation algorithm to obtain a sub-optimal solution in Section 6.4;

(iii) to consider online requests, we present our online algorithm in Section 6.5.

6.3 Optimal Solution

In the literature, the optimal solution for mTSP is given by Integer Linear Programming. Thus, we formulate our Carpool Route Calculation with following definitions.

- (1) $G = (V, A)$: a weighted complete graph where vertex a is the origin vertex where a carpool starts and $V' = V - \{a\}$ is the set for destinations, and a weight on A indicated as c_{ij} is the real world mileage cost from vertex i to vertex j ;
- (2) $x_{ij} = 1$ if edge $(i, j) \in A$ is used; $x_{ij} = 0$ otherwise;
- (3) $[t_i^s, t_i^e]$: a travel period for a passenger to vertex i ;
- (4) n : the number of available taxicabs;
- (5) c : the taxicab capacity;
- (6) y_i : total number of dropped passengers before vertex i ;
- (7) q_i : total number of dropped passengers at vertex i ;
- (8) p_i : time arriving at vertex i ;
- (9) w_i : latest starttime of dropped passengers before vertex i ;
- (10) $T(i, j)$: travel time between vertex i and vertex j .

$$\begin{aligned}
 \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 \text{s.t.} \quad & \sum_{i \in V'} x_{ij} = 1 \quad \forall j \in V' \quad (1) \\
 & \sum_{j \in V'} x_{ij} \in \{0, 1\} \quad \forall i \in V' \quad (2) \\
 & \sum_{j \in V'} x_{aj} \leq n \quad \sum_{i \in V'} x_{ia} = 0 \quad (3) \\
 & y_i + q_i \leq c \quad \forall i \in V' \quad (4) \\
 & \text{If } x_{ij} = 1 \Rightarrow y_i + q_i \leq y_j \quad \forall i, j \in V' \quad (5) \\
 & p_i \leq t_i^e \quad \forall i \in V' \quad (6) \\
 & \text{If } x_{ij} = 1 \Rightarrow p_i + T(i, j) \leq p_j \quad \forall i, j \in V' \quad (7) \\
 & \max\{w_i, t_i^s\} + T(a, i) \leq t_i^e \quad \forall i \in V' \quad (8) \\
 & \text{If } x_{ij} = 1 \Rightarrow w_i \leq w_j \quad \forall i, j \in V' \quad (9) \\
 & \sum_{i \notin S} \sum_{j \in S} x_{ij} \geq 1 \quad \forall S \subseteq V' \quad (10) \\
 & x_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad (11)
 \end{aligned}$$

where (1) ensures that exactly one taxicab visits a destination; (2) ensures that exactly one taxicab leaves one destination for the next delivery, or the carpool is over and no delivery needs to be made; (3) is about the constraint on the number of available taxicabs; (4) and (5) are about the taxicab capacity constraint; (6), (7), (8) and (9) are about the travel period constraint; (10) is to prevent the formation of subtours not including origin vertex a . Note that although every taxicab has disjoint vertices in a delivery graph, they can share the same route in the real world when performing the carpool.

Since the traditional traveling salesmen problem is NP-Hard, as a generalized version, our problem is also NP-Hard (due to space constraint we omit the formal proof). Therefore, when the number of destinations increases, the running time to solve the above integer programming increases exponentially. Although integer linear programming is sufficient for a small number of destinations, we need to accommodate the case where the number of destinations is large with an efficient algorithm.

6.4 Approximation Algorithm

In this section, we first propose an approximation algorithm, and discuss the impacts of the three constraints.

6.4.1 Approximation Algorithm without Constraints

An approximation algorithm without constraints is under a scenario where all passengers' travel periods are not considered; the origin is with unlimited taxicabs for carpooling; every taxicab has unlimited capacities.

We first present some rationales. Any carpool from the same origin can be performed with two key steps: (i) we shall *assign passengers* to different empty taxicabs; (ii) we shall calculate a *delivery order* for a given passenger assignment for a particular taxicab. In the followings, we will describe how to make use of this rationale for our approximation algorithm in two steps and provide its approximation ratio.

(i) **Passenger Assignment:** Based on a complete graph G created with delivery requests, we will show how to assign passengers in Figure 13.

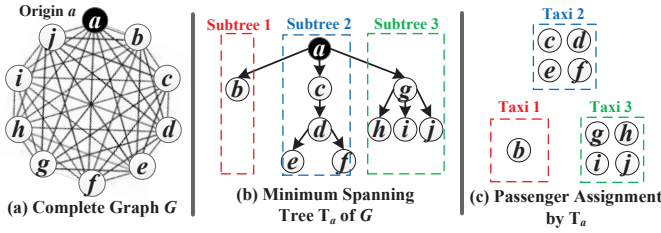


Fig 13. Passenger Assignment

To assign passengers to different taxicabs, we shall take into account the distances between destinations given by G in subgraph (a). The objective is to find a *minimum weight* subgraph of G to assign destination vertices to different paths (every path is used by a unique taxicab). Since the minimum spanning tree (MST) is the minimum weight subgraph of G , in this paper we try to employ an MST to obtain a passenger assignment. Subgraph (b) gives a G 's MST T_a with three subtrees rooted at origin vertex a . Based T_a , we assign the passengers, who have destination vertices in the same subtree, into the same taxicab. For example, passengers with destinations c, d, e and f are assigned into Taxicab 2, as in Subgraph (c). Note that T_a is not a delivery graph we try to obtain, because T_a has branches at destination vertices. Thus, we have shown how to conduct passenger assignment based on a given complete graph.

(ii) **Delivery Order Calculation:** Based on the passenger assignment in step (i), in Figure 14 we will show how to calculate a delivery order among passengers assigned into the same taxicab.

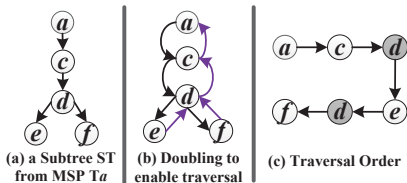


Fig 14. Delivery Order Calculation

As in step (i), a passenger assignment for a particular taxicab is given by a subtree rooted at the origin vertex a . Thus,

we employ Subtree 2 (named ST) in the MST T_a in subfigure (a) as an example to show how to obtain a delivery order. With an observation on ST , we found that ST only gives a passenger assignment, but not a fixed delivery order since ST has a branch that requires passenger relay, which is prohibited by *coRide*. Thus, a new subgraph transformed from ST should be created to calculate an order without relay. In this paper, we use a depth-first traversal from root vertex to decide a delivery order. But as we can see in subfigure (a), ST is a directed graph, and cannot be traversed based on current edges. Thus, as in subfigure (b), we double the edges in ST to create loops to enable a traversal $a \rightarrow c \rightarrow d \rightarrow e \rightarrow d \rightarrow f$. This order is not a delivery order since it involves duplicated vertices, *i.e.*, d , thus a longer total mileage $M_{ac} + M_{cd} + M_{de} + M_{ed} + M_{df}$.

To obtain a delivery order, we use a shortcut strategy to eliminate duplicated vertices in a traversal. In Figure 15, we show how to shortcut some edges about duplicated vertices, thus further reducing a delivery mileage.

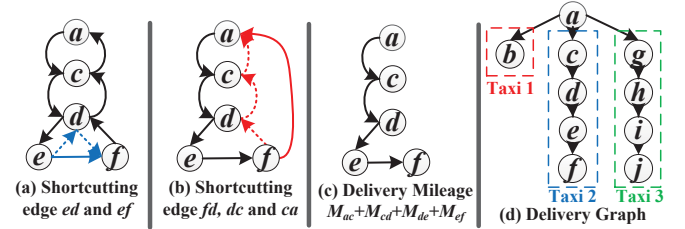


Fig 15. Shortcutting about Duplicated Vertices

As in subfigure (a) and (b), we shortcut edge ed and df with a new edge ef , and then shortcut edge fd, dc and ca with another new edge fa . Note that based on triangle inequality, the length of an added edge (*e.g.*, M_{ef}) is always shorter than the sum of edges it shortcutting (*e.g.*, $M_{ed} + M_{df}$). Further, we delete the edge fa to obtain the delivery order $a \rightarrow c \rightarrow d \rightarrow e \rightarrow f$ and the total mileage cost of 4 edges ($M_{ac} + M_{cd} + M_{de} + M_{ef}$) as in subfigure (c). Therefore, with a traversal, we have shown that how to calculate a delivery order based on a given passenger assignment. With the above two steps, we finish our approximation algorithm to obtain our delivery graph in subfigure (d).

Proof of Approximation Ratio: We have proved that our traversal algorithm has a constant performance ratio of 2, *i.e.*, the total mileage obtained by our carpool schedule, is at most 2 times the optimal mileage we obtained by the optimal solution using integer programming. This is because (i) with shortcutting, the weight of our delivery graph $W(S)$ is smaller than a weight of a Traversal $W(T')$, *i.e.*, $W(S) < W(T')$; (ii) a traversal is exactly two times of a MST, $W(T') = 2W(T)$; (iii) the MST is smaller than or equal to the Optimal solution since the optimal solution is a spanning tree and MST is the smallest spanning tree, $2W(T) \leq 2W(O)$. Thus, $W(S) < W(T') = 2W(T) \leq 2W(O)$, therefore $\frac{W(S)}{W(O)} < 2$.

During the construction of the minimum spanning tree, three constraints, *i.e.*, Taxicab Capacity, Number of Available Taxicabs, Travel Period, have special impacts, which will be introduced in the following three subsections.

6.4.2 Impact of Number of Available Taxicabs n

In this section, we show how to solve a carpool problem with constraints on the number of available taxicabs n .

We can reduce the total mileage by delivering passengers separately, if they are heading in significantly different directions. In a delivery graph G , every subtree rooted at the origin is associated with a separate taxicab, which satisfies all delivery requests in this subtree. For example, in Figure 16, the spanning tree has three subtrees (boxed), and therefore we need three taxicabs to satisfy the deliveries.

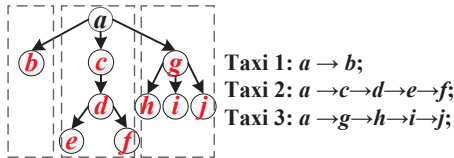


Fig 16. Minimum Number of Taxicabs for Deliveries

When constructing a spanning tree, we have to find one spanning tree whose number of subtrees rooted at origin is not bigger than n . Figure 17 shows how to impose such a constraint during a spanning tree construction.

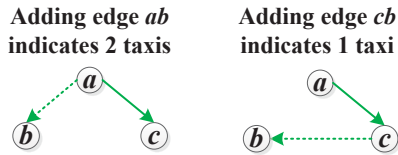


Fig 17. Constraints on Number of Available Taxicabs n

In Figure 17, given $n = 1$, *i.e.*, there is only one taxicab available for origin a , suppose that after adding edge ac , currently the minimum edge that should be added to the tree is edge ab according to Prim's algorithm. But adding edge ab indicates that we need two taxicabs to fulfill the deliveries, which is against to $n = 1$. Alternatively, we can add edge cb and it will still fulfill the deliveries yet with one taxicab.

6.4.3 Impact of Taxicab Capacity c

In this section, we consider how to solve a carpool problem with constraints on the taxicab capacity c .

Since the taxicab capacity is limited (*e.g.*, 4 for a sedan and 6 for a van), a delivery graph should not have infinite depth for any delivery branch. It is clear that given a fixed spanning tree, the minimum taxicab capacity is equal to the size of its biggest subtree rooted at origin, because a taxicab has to deliver all passengers in this subtree from origin. Figure 18 gives an example, where the biggest subtree has four vertex, therefore the minimum taxicab capacity is 4.

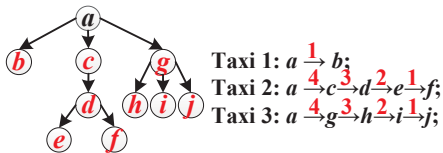


Fig 18. Minimum Taxicab Capacity for Deliveries

When constructing a spanning tree, we have to control the sizes of subtrees to make sure the size of the largest subtree

less than given capacity constraint c . Figure 19 shows how to consider it during the construction of a spanning tree.

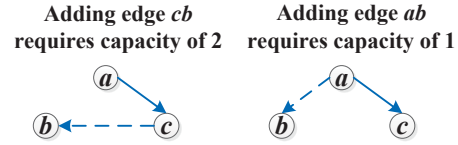


Fig 19. Constraints on Taxicab Capacity c

In Figure 19, suppose $c = 1$ for simplicity, and suppose that after adding edge ac , currently the minimum edge should be added to the tree is edge cb according to Prim's algorithm. But adding edge cb indicates that we need taxicabs with capacity of 2 to fulfill the deliveries, which is against to $c = 1$. Alternatively, we can add edge ab and it will still fulfill the deliveries yet with capacity of 1.

6.4.4 Impact of Travel Period

In this subsection, we analyze the carpool problem with constraints on the travel periods of deliveries.

A travel period of a delivery i is specified by $[t_i^s, t_i^e]$, where t_i^s is the earliest time that a delivery i can start, and t_i^e is the latest time that delivery i must finish. The reason to consider travel periods is that in practice, two deliveries with non-overlapping periods cannot be carpoled together, even though they have the same origin and destination.

To impose this constraint, for a minimum spanning tree T_a and a delivery i with travel period $[t_i^s, t_i^e]$ from origin vertex a to destination vertex i , we need to ensure that a spanning tree T_a can accommodate delivery i by satisfying:

$$\max_{k \in p} t_k^s + T(a, i) \leq t_i^e,$$

where p is a path on T_a from a to i , hence $\max_{k \in p} t_k^s$ is the start time of last passenger, and $T(a, i)$ is the travel time from a to i in p of T_a . The left-hand side is expected arrival time of delivery i according to this carpool, and the right-hand side is the latest end time of delivery i , given by the passenger. Thus, if the left-hand side is smaller than or equal to the right-hand side, it indicates that T_a can accommodate i . Figure 20 gives an example of how to validate whether the minimum spanning tree can accommodate a delivery or not.

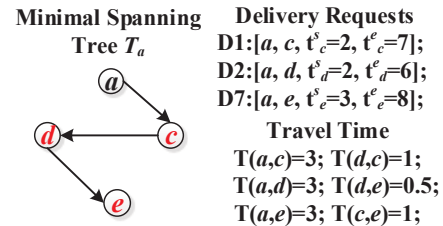


Fig 20. Validation on Travel Period

In Figure 20, suppose that during the construction of a spanning tree, the next minimum edge should be added to the spanning tree according to Prim's algorithm is an edge de . Based on delivery requests and travel time in Figure 20, $\max_{k \in p} t_k^s = \max\{2, 2, 3\} = 3$; $T(a, e) = 3 + 1 + 0.5 = 4.5$; thus, $\max_{k \in p = \{a \rightarrow c \rightarrow d \rightarrow e\}} t_k^s + T(a, e) = 7.5 \leq t_e^e = 8$. Therefore, the edge de is a safe edge and can be added to the spanning tree.

6.4.5 Put All Constraints Together

A practical approximation algorithm shall construct a minimum spanning tree that (i) accommodates all travel periods of its deliveries, (ii) has the biggest size of subtrees not bigger than c , and (iii) has a number of subtrees not bigger than n . The details about how to impose the three constraints have been given in pervious section. We note that the order of imposing constraints should not be changed, since it is easiest to find more taxicabs to fulfill requests, relatively easier to find bigger taxicabs to fulfill requests, and harder to require passengers to change their schedules. If the conditions conflict with each other, we can always find a feasible solution by using more taxicabs.

Note that with highly diverse travel periods or a small taxicab capacity, lots of taxicabs will be used to satisfy deliveries individually, which is a delivery schedule based on “fat” spanning trees with small yet many subtrees. In contrast, with a small number of available taxicabs, lots of deliveries will be pooled into one taxicab and then be fulfilled one by one, which is a delivery schedule based on “thin” spanning trees with big yet fewer subtrees.

6.5 Algorithm for Online Requests

Instead of providing delivery requests a day or two before the delivery start time, some passengers may call a dispatching center a hour, or even several minutes, before the delivery start time to provide online requests. In *coRide*, we response to online requests by adding them to an existing carpool schedule, or start a new carpool schedule based on details of requests.

Given an online request k and a delivery graph DG about existing carpool route schedules at an origin, we can integrate k to DG by two methods:

(i) **As a new leaf to an existing subtree** of DG : we add k to the leaf of a subtree of DG (k as the new and the only leaf for this subtree), only if (a) this new subtree can accommodate the travel period of k and (b) the size of this new subtree is not bigger than the taxicab capacity c . This method indicates that we pool this new passenger with passengers of an existing carpool.

(ii) **As a new subtree** of DG : we add k to DG by constructing a new subtree (k as the only one vertex in this subtree) from the origin vertex of DG . This method indicates that we start a new separate carpool for this new passenger.

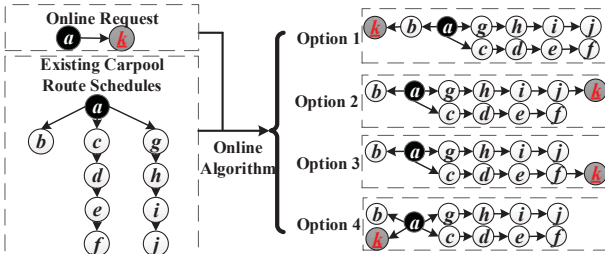


Fig 21. Online Algorithm for Online Requests

In Figure 21, given the existing carpool route schedules and an online request, *coRide* obtains four different carpool options for the passenger to select by pooling this passenger to existing taxicabs (option 1, 2 and 3) or starting a new carpool (option 4). The key criteria for passengers to select a carpool is the carpool fare as we present next.

7 The Win-Win Fare Model

Generally, a taxicab fare consists of three main parts: an initial charge for every service; surcharge for luggage, waiting time, *etc*; and main charge based on traveled distance. In our model, we focus on how to consider a carpool benefit into calculations of the main charge. Such a carpool benefit shall be shared between the passengers (as a group) and the driver, as well as among the passengers themselves. The rationale behind sharing the carpool benefits with drivers is that we have to encourage drivers to participate in the non-mandatory carpool application. We believe that negotiating privately by passengers alone and sharing the benefits only between passengers will severely hurt the interests of drivers, since the total profit for all drivers will decrease significantly.

7.1 Carpool Benefit

A carpool benefit \mathbf{B} between the total non-carpool fare and a fare paid for a carpool distance is given as follows:

$$\mathbf{B} = \sum_{i=1}^c \tau_i - \tau,$$

where c is the total number of passengers in this carpool; τ_i is the separate non-carpool fare for passenger i ; τ is the regular fare for a distance equal to the carpool distance (not the carpool fare). Thus, the total non-carpool fare of all passengers is given by $\sum \tau_i$, and the regular fare for the carpool distance is given by τ , and their difference is a carpool benefit \mathbf{B} . Given a carpool schedule, all three parameters are obtainable, and thus \mathbf{B} is also obtainable.

For example, Figure 22 shows three passengers (with non-carpool fare $\tau_1 = 17$, $\tau_2 = 32$, $\tau_3 = 45$) carpooled together with a distance of a regular fare $\tau = 52$, leading to $\mathbf{B} = 42$. Note that $\tau = 52$ is a regular fare for a distance equal to the carpool distance, and is not the actual carpool fare all passengers will pay together under our model.

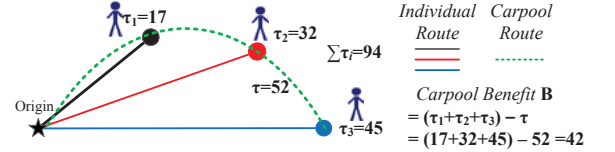


Fig 22. Carpool Benefit

To build a win-win fare model, we need to (i) share a carpool benefit between the driver and all passengers as a group and (ii) share the benefit within the passenger group.

7.2 Sharing % between Driver & Passenger

We use ρ to indicate the sharing percentage of the passengers (all passengers as a group) for a given carpool benefit \mathbf{B} , and hence $1 - \rho$ is the sharing percentage of the driver.

(i) For a carpool benefit \mathbf{B} , all passengers as a group pay:

$$\text{Total Fare Paid by Passengers} = \sum_{i=1}^c \tau_i - \rho \times \mathbf{B},$$

where $\sum \tau_i$ is the sum of regular fares by all passengers in a non-carpool situation; $\rho \times \mathbf{B}$ is the benefit to passenger group.

(ii) For a carpool benefit \mathbf{B} , a driver collects:

$$\text{Total Fare collected by Drivers} = \tau + (1 - \rho) \times \mathbf{B},$$

where τ is the fare a driver collects for the carpool distance; $(1 - \rho) \times \mathbf{B}$ is the benefit for a driver to carpool. Note it is easy to check that the total carpool fare paid by passengers equals the amount collected by the driver in our model.

In real world scenarios, ρ can be dynamically decided based on various factors about the supply and request relationship in a taxicab network. In this paper, we give an example to define $\rho = \frac{\# \text{ of occupied taxicabs}}{\# \text{ of total taxicabs}}$ in a certain area during a time window to balance the carpool incentives between the driver and the passenger. Thus, for a large ρ , *i.e.*, more occupied taxicabs, the more benefit will be given to the passengers to encourage passengers to carpool; for a small ρ , *i.e.*, more empty taxicabs, the more benefit will be given to the driver to discourage passengers to carpool, balancing deliveries among other empty taxicabs. In Figure 22, given $\rho = \frac{1}{2}$, total carpool fare collected by drivers is $52 + \frac{1}{2} \times 42 = 73$, which is equal to the total carpool fare for all passengers, *i.e.*, $94 - \frac{1}{2} \times 42 = 73$.

7.3 Sharing % among Passengers

Among the total carpool benefits for all passengers, *i.e.*, $\rho \times \mathbf{B}$, we shall decide a sharing percentage to show a carpool benefit for a particular passenger i , and thus model the carpool fare for a passenger i . It is given as follows.

$$\text{Carpool Fare Paid by a Passenger } i = \tau_i - \rho \times \mathbf{B} \times \frac{\tau_i}{\sum \tau_i},$$

where τ_i is the non-carpool fare a passenger i has to pay at a non-carpool situation; $\rho \times \mathbf{B} \times \frac{\tau_i}{\sum \tau_i}$ is the carpool benefit for a particular passenger i . In Figure 22, given $\rho = \frac{1}{2}$, the carpool fare paid by a passenger 3 is $45 - \frac{1}{2} \times 42 \times \frac{45}{94} \approx 34$.

Currently, we use $\frac{\tau_i}{\sum \tau_i}$ to share the carpool benefit among passengers based on their non-carpool fare. In other words, we differentiate passengers by their destinations to the common origin, not the delivery order. But the last dropped off passenger typically will have a farther destination than other passengers (since our carpool graph is based on the minimum spanning tree), so he/she will share more carpool benefit than other earlier dropped off passengers in our fare model, which implicitly compensates to the passengers with a longer traveling time. In more advanced designs, the sharing percentages among passengers can also be directly decided by the priority of services, *e.g.*, based on the delivery order μ_i of passenger i in a carpool, the sharing percentage can be defined as $\frac{\mu_i}{\sum \mu_j}$.

7.4 Carpool Fare Model Evaluation

In this subsection, we numerically evaluate our fare model. Based on three delivery requests in Figure 22, Figure 23 shows the impact of different sharing percentages ρ on the fare that every passenger paid and the fare the driver collects. It shows when ρ increases from 0 to 1 (indicating a trend of an undersupplied taxicab services in the real world), the carpool incentive for the passenger increases from 0% fare savings to 44% fare savings, whereas the carpool incentive for the driver decreases from 80% more profit to 0% more

profit. By adjusting sharing percentage ρ according to taxicab supply, our model can dynamically balance the carpool incentives for drivers and passengers.

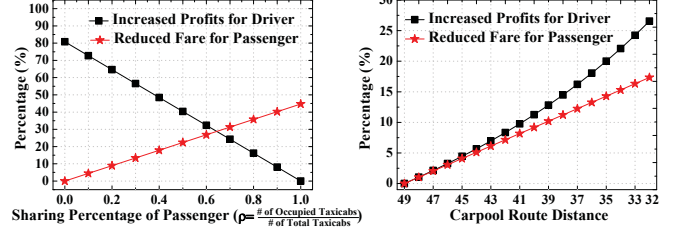


Fig 23. Incentive Balancing Fig 24. Win-Win Model

Given requests with fixed non-carpool fares, a short carpool distance will increase the carpool benefit (the same $\sum \tau_i$, but a smaller τ), which results in a win-win situation (*i.e.*, more profits for drivers and lower fares for passengers). Taking the passengers with $\tau_1 = 17$ and $\tau_2 = 32$ in Figure 22 as examples, physically, the lower bound of a fare paid for the carpool distances should be $\tau_{min} = \max\{\tau_1, \tau_2\} = 32$. In addition, passengers may not select a carpool delivery where they pay a fare together more than the sum of their regular non-carpool fares. So, logically, the upper bound for τ is $\tau_{max} = \tau_1 + \tau_2 = 49$. Figure 24 show impacts of different carpool route distances (by different τ from τ_{max} to τ_{min}) on the savings percentages of passengers and profiting percentages of drivers. First, it shows a win-win situation as long as $\tau < \tau_{max}$. Second, the smaller τ , the higher the profit for drivers, the lower the fare for passengers.

8 Evaluation of the *coRide* Service

We have installed the customized TaxiBox in a small portion (98 taxicabs) of the taxicab network of a Chinese city Shenzhen with a population of 10 million to test the functionality of TaxiBox. We quickly learned that it takes time to install hardware in current taxicabs, and that it is much more difficult than we had anticipated. Although the taxicab operators requested that their drivers cooperate with the deployment, drivers still were not enthusiastic about installing devices to taxicabs with no immediate benefits to them. During the deployment, it was usual for drivers to not appear or to arrive late and leave early due to business or personal matters. It was also hard to persuade drivers to be more involved in system testing, *e.g.*, logging passenger numbers for every delivery. How to provide an incentive for them to be involved in system deployment and testing is a key question we need to address.

For a large scale deployment of carpool services, through the operators from which we obtained datasets, the dispatching center to collect delivery requests via phone call has been established. But the detailed regulation law about taxicab carpool are still under progress to being passed, and hope to be completed within this year. Thus, a large scale carpool service evaluation is hard to conduct for current situation. Instead, we perform a large scale trace-driven evaluation of a real world dataset about one week of GPS records of 14,453 taxicabs belonging to different taxicab companies in Shenzhen.

8.1 Datasets

The first dataset contains daily GPS trace data of the taxicab network, and the second dataset is about deliveries. The GPS dataset was collected by letting each taxicab upload its records 30 seconds on average to a centralized base station, and the delivery dataset was obtained by an offline method. Figure 25 gives details about these datasets.

Description of Datasets			
GPS Dataset		Delivery Dataset	
Collection Period	01/01/12-06/30/12	Collection Period	01/01/12-06/30/12
Numbe of Taxis	14,453	Numbe of Taxis	14,453
Data Size	450GB	Data Size	18GB
Record Number	3,888,000,000	Record Number	95,000,000
Format		Format	
Plate Number	Date and Time	Plate Number	Begin & End Time
Status	Speed	Delivery Distance	Delivery Duration
Direction	GPS Coordinates	Delivery Fare	Unload Distance

Fig 25. Details of Datasets

In the GPS dataset, key attributes are taxicab status and GPS coordinates, which can indicate whether a taxicab at a certain location is empty or not. In delivery dataset, the key attributes are delivery distance, duration and fare, which can describe a taxicab delivery event. Further, the unload distance indicates the distance between the end location of the last delivery and the begin location of this delivery. By combining these two datasets, we can fully understand the daily operational situation of the entire taxicab network and conduct a valid evaluation. Due to the large size of the datasets, we mainly found two kinds of errors. (i) Location Error: GPS coordinates show that a taxicab is off the road. (ii) Missing Records: a fair amount of GPS records are missing. The errors may result from different reasons, *e.g.*, GPS device malfunctions, software issues, *etc.* We perform a preprocessing to clean datasets to rule out taxicabs with more than 10% of missing or errant records.

Due to security and privacy reasons, we are allowed to select one week of trace data from the half year dataset as a sample for evaluation purposes.

8.2 Evaluation Overview

To show the effectiveness of carpool services, we compare two carpool route calculation algorithms, the *optimal carpool* and the approximation algorithm, indicated as *coRide*, with the *ground truth*, which is the original GPS traces from the dataset. To show the performance of *coRide* to address online requests, we also plot the performance of *coRide online*.

The above algorithms are evaluated based on three different real world constraints. (i) **Taxicab Capacity** c to show that how many deliveries can be pooled together in a single taxicab. (ii) **Number of Available Taxicabs** n to show that how many taxicabs can be used at an origin to fulfill all delivery requests. (iii) **Travel Period** $[t_i^s, t_i^e]$ to show the delivery start time and a tolerated end time. For travel period constraints, since we can obtain the actual travel period about every delivery in the dataset, we use a *tolerated detour time* t (minutes) plus the actual end time of a trip to show this constrain. For example, for an actual travel period $[t_i^s, t_i^e]$ in the dataset about delivery i , with a tolerated detour time t , the travel period we used to test a spanning tree is $[t_i^s, t_i^e + t]$, instead of the actual travel period $[t_i^s, t_i^e]$.

From the three perspectives of society, passengers, and drivers, we evaluate the performance of the above algorithms by several metrics. From society's perspective, with the **Percentage of Reduced Total Mileage**, we investigate how much mileage we can reduce by carpooling, given the above constraints and different time lengths between the time to provide delivery requests and time to start deliveries for online requests. From passengers' perspective, with the **Percentage of Reduced Fare** paid by passengers, we show the minimum fare they can pay, given tolerated detour times. From drivers' perspective, with the **Percentage of Increased Profit** earned by drivers, we present the maximum fare they can collect, given tolerated detour times. In addition, we also investigate two practical metrics, *i.e.*, (i) the running time of the optimal algorithm to show why this optimal algorithm is not feasible in terms of running time, and (2) the increased individual mileage due to carpooling to show a possible negative effect of carpooling, *i.e.*, increasing the travel time for passengers.

In the evaluation, for datasets about individual days of the week, we first process datasets to obtain delivery requests, and then based on the delivery requests we calculate the carpool route by different algorithms. By processing these requests on a daily basis, we show the performance when passengers provide delivery requests 24 hours earlier than the delivery start time, and for requests starting at one day and ending at the day after, we classify them into the day they start. For *coRide* online, we show its performance when passengers provide requests at 1, 3, 6 and 12 hours earlier than the delivery start time. The results are average outcomes of 7 days of evaluations.

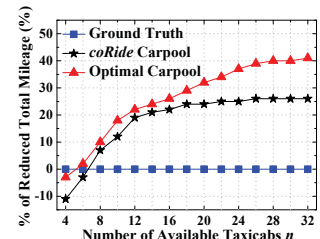
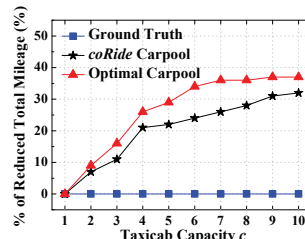


Fig 26. Total Mileage vs. c Fig 27. Total Mileage vs. n

8.3 Reduced Total Mileage

In this subsection, we evaluate *coRide* via the percentage of reduced total mileage at different parameters.

8.3.1 Taxicab Capacity c

Figure 26 plots the effect of taxicab capacity c on the percentage of reduced total mileage with tolerated detour time $t = 5$ and number of available taxicabs $n = 16$. With the increase of taxicab capacity c , the percentage of reduced total mileage for *coRide* carpool and the optimal carpool also increases. For example, in *coRide* carpool, the percentage of reduced total mileage increases from 0% to 22%, when taxicab capacity c increases from 1 to 4. This is because when taxicab capacity c increases, a delivery of a taxicab can be pooled with more other deliveries, and thus it can reduce the total mileage. It implies that a carpool functions more effectively when taxicabs can carry more passengers.

8.3.2 Number of Available Taxicabs n

Figure 27 plots the effect of the different number of available taxicabs n on the percentage of reduced total mileage with tolerated detour time $t = 5$ and taxicab capacity $c = 4$. We observe that with the increase of number of available taxicabs n , the percentages of reduced total mileage in *coRide* carpool increase from -11% to 27% . These are some negative percentages of reduced total mileage when the number of available taxicabs n is small, and a similar situation is also shown in the performance of the optimal carpool. This is because that with fewer taxicabs at an origin, we have to pool more unrelated deliveries in this origin into the same taxicab, and drop them off one by one, and it will increase the total mileage. But when the number of available taxicabs n is larger than 8, we can reduce the total mileage by carpools.

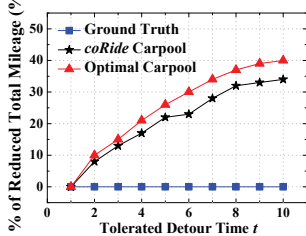


Fig 28. Total Mileage vs. t

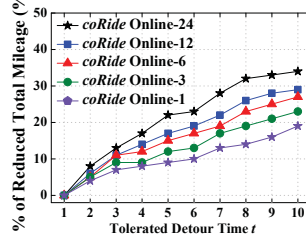


Fig 29. Mileage (Online)

8.3.3 Travel Period $[t_i^s, t_i^e]$

Figure 28 plots the effect of different travel periods in terms of different tolerated detour time on the percentage of reduced total mileage with the number of available taxicabs $n = 16$ and taxicab capacity $c = 4$. In Figure 28, we observe that with the increase of tolerated detour time t in terms of minutes, the percentages of reduced total mileage in *coRide* carpool increase from 0% to 33% , while these of the optimal carpool increase from 0% to 40% , leading to a 7% performance gain. While in a carpool, with more detour time, more mileage can be reduced by pooling more deliveries together. The increase of t enables a larger travel period, making more deliveries correlated with each other in time.

8.3.4 Online Requests

In the above *coRide* carpool, we process requests by days, so it means we pool the delivery requests that passengers provided by 24 hours in advance (named *coRide* online-24). In Figure 29, we evaluate the performance of *coRide* for online requests situations where (i) the half of the passengers provides requests in advance of 24 hours, and based on them, we build carpool graphs, (ii) the other half of the passengers provides requests in advance of 1, 3, 6 and 12 hours (indicated as *coRide* online-1, etc), and we use our online algorithm to optimally add these online requests together to the existing carpool graphs every 1, 3, 6 or 12 hours, leading to new different carpool graphs. We observe that *coRide* online-24 outperforms all others, indicating the early the passengers provide requests, the better the performance. This is because with more requests to begin with, we can build a more effective spanning tree, but for online requests, we only can add requests as leaves or start a new branch at roots of carpool graphs.

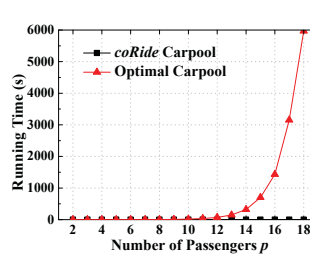


Fig 30. Time vs. p

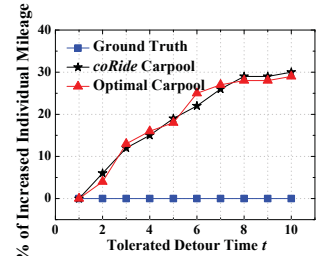


Fig 31. Mileage vs. t

8.3.5 Running Time of Algorithms

Figure 30 shows the running time of the optimal carpool algorithm and *coRide* carpool algorithm at different carpool passenger numbers p at a single origin. We observe that as the passenger number p increases from 2 to 18, the running time for *coRide* carpool algorithm is negligible compared to the running time for optimal carpool algorithm. This is because that our carpool route calculation problem is NP-hard, and the optimal carpool algorithm uses Integer Programming to obtain the solution, which leads to a longer running time, and is not practical for real world carpool route calculation with a large number of passengers.

8.3.6 Percentage of Increased Individual Mileage

In this subsection, we evaluate the performance of *coRide* carpool by the percentage of increased individual mileage due to carpools with different travel periods. This increased individual mileage also provides an indication of the detour time a passenger will tolerate for carpooling with others. Note that although the individual mileage increases, the fare for individual passengers is actually reduced, since more passengers will share the fare for common routes, leading to a large carpool benefit, as showed by our Fare Model in Section 7. Figure 31 plots the effect of different travel periods in terms of t on the percentage of increased individual mileage with $n = 16$ and $c = 4$. With the increase of t from 1 to 10, the percentage of increased individual mileage in *coRide* carpool increases from 0% to 30% , while that of the optimal carpool has a similar trend. In *coRide* and the optimal carpool, with more detour time, a high mileage is added to individual deliveries, since after carpool, most of the passengers will have a new yet longer route compared to the ground truth.

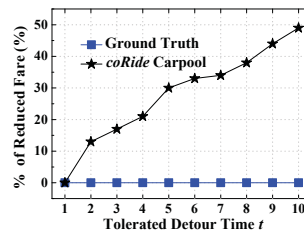


Fig 32. Reduced Fare

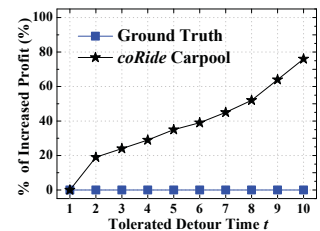


Fig 33. Increased Profit

8.4 Reduced Fare for Passengers

In this subsection, we evaluate the performance of *coRide* carpool in terms of maximally reducing the fare for individual passengers, based on the win-win fare model we proposed in Section 7. Based on the datasets, we have the ground truth for regular fares of individual passengers, and

based on the carpool route, we shall have the carpool fare. We let all the passengers and the driver to evenly share the carpool benefit due to the mileage reduction of a carpool route. In Figure 32, we observe that with the increase of tolerated detour time t , the percentages of reduced fare for individual passengers in *coRide* carpool increase from 0% to as much as 49%. In a carpool, with more detour time, high mileage can be shared with other passengers, thus leading to a large carpool benefit for fare reductions. It will lead to an economic incentive for passengers to carpool.

8.5 Increased Profit for Drivers

In this subsection, we evaluate the performance of *coRide* carpool in terms of maximally increasing the profit for taxicab drivers based on our win-win fare model. With the method similar to that of the last subsection, we can produce an increased profit by comparing the total carpool fare collected by the taxicab driver, and the ground truth of the regular fare about the first passenger picked up in the carpool, which gives the fare the driver will collect in the case that no carpool is conducted. In Figure 33, we plot the effects of different travel periods in terms of t on the percentage of increased benefits. It shows that with the increase of tolerated detour time t , the percentage of increased benefits for the drivers in *coRide* carpool increases from 0% to as much as 76%, which leads to a considerable incentive for taxicab drivers to take carpool trips.

9 Conclusion

In this work, we analyze, design, implement, and evaluate a prototype taxicab carpool system *coRide* to reduce the total mileage to deliver passengers. Our effort provides a few valuable insights and guidelines, which are hoped to be useful for realizing carpooling services commercially in near future. Specifically, (i) we found unprecedented evidence of inefficiencies of current systems, and opportunities for new systems based on our real-world datasets; (ii) we implement a customized hardware supporting the essential functionalities for carpooling; (iii) we affirmed that complicated route functions should be implemented in a centralized cloud and near optimality can be achieved; (iv) it is important to establish incentives for all the parties involved (e.g., a win-win situation); and (v) finally our work only addresses the technical frontier, and it is even more critical to establish a right policy that would make a large scale deployment feasible.

10 Acknowledgements

The authors thank our shepherd Dr. Jie Liu and the anonymous reviewers for their valuable feedbacks. This work was supported in part by the US NSF Grants CNS-0845994, CNS-0917097, CNS-1239226, IBM OCR Fund, and by the China NSFC Grants 60903222, 61100220, 61170247 and 61322205, as well as China National Major Science and Technology Project 2013ZX01033002-003.

11 References

- [1] Supershuttle service. <http://www.supershuttle.com/>.
- [2] Taxi user surveys. San Francisco Municipal Transportation Agency.
- [3] ASLAM, J., LIM, S., PAN, X., AND RUS, D. City-scale traffic estimation from a roving sensor network. In *Proceedings of 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12.
- [4] BALAN, R. K., NGUYEN, K. X., AND JIANG, L. Real-time trip information service for a large taxi fleet. In *Proceedings of the international conference on Mobile systems, applications, and services*, MobiSys '11.
- [5] BIAGIONI, J., AND ERIKSSON, J. Map inference in the face of noise and disparity. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '12.
- [6] CONSULTING, S. The new york city taxicab fact book. <http://www.schallerconsult.com/taxi/taxifb.pdf>.
- [7] DATA100 COMPANY. *Taxicab Carpooling Survey*, 2012. <http://wenku.baidu.com/view/2f0fea1f964bcf84b9d57b4c.html>.
- [8] GE, Y., LIU, C., XIONG, H., AND CHEN, J. A taxi business intelligence system. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11.
- [9] GE, Y., XIONG, H., TUZHILIN, A., XIAO, K., GRUTESER, M., AND PAZZANI, M. An energy-efficient mobile recommender system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), KDD '10.
- [10] HUANG, Y., AND POWELL, J. W. Detecting regions of disequilibrium in taxi services under uncertainty. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (2012), SIGSPATIAL '12.
- [11] LIU, W., ZHENG, Y., CHAWLA, S., YUAN, J., AND XING, X. Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (2011), KDD '11.
- [12] LIU, X., BIAGIONI, J., ERIKSSON, J., WANG, Y., FORMAN, G., AND ZHU, Y. Mining large-scale, sparse gps traces for map inference: comparison of approaches. In *Proceedings of the international conference on Knowledge discovery and data mining*, KDD '12.
- [13] NATIONMASTER. Energy statistics. http://www.nationmaster.com/graph/ene_oil_con-energy-oil-consumption.
- [14] OBERLIN, P., RATHINAM, S., AND DARBHA, S. A transformation for a multiple depot, multiple traveling salesman problem. In *Proceedings of the conference on American Control Conference*, ACC'09.
- [15] TAXI, N., AND COMMISSION, L. Taxi of tomorrow survey.
- [16] TIMES, N. Y. Limited share-a-cab test to begin soon. <http://www.nytimes.com/2010/02/22/nyregion/22ataxis.html>.
- [17] WEI, L.-Y., ZHENG, Y., AND PENG, W.-C. Constructing popular routes from uncertain trajectories. In *Proceedings of the international conference on Knowledge discovery and data mining*, KDD '12.
- [18] WIKIPEDIA. Emissions trading or cap and trade. <http://en.wikipedia.org/wiki/Emissiontrading>.
- [19] WU, W., NG, W. S., KRISHNASWAMY, S., AND SINHA, A. To taxi or not to taxi? - enabling personalised and real-time transportation decisions for mobile users. In *Proceedings of the 2012 IEEE 13th International Conference on Mobile Data Management*, MDM '12.
- [20] YUAN, J., ZHENG, Y., AND XIE, X. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (2012), KDD '12.
- [21] YUAN, J., ZHENG, Y., XIE, X., AND SUN, G. Driving with knowledge from the physical world. In *Proceedings of the international conference on Knowledge discovery and data mining*, KDD '11.
- [22] YUAN, J., ZHENG, Y., ZHANG, C., XIE, W., XIE, X., SUN, G., AND HUANG, Y. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10.
- [23] YUAN, J., ZHENG, Y., ZHANG, L., XIE, X., AND SUN, G. Where to find my next passenger. In *Proceedings of the 13th international conference on Ubiquitous computing* (2011), UbiComp '11.
- [24] ZHANG, D., LI, N., ZHOU, Z.-H., CHEN, C., SUN, L., AND LI, S. ibat: detecting anomalous taxi trajectories from gps traces. In *13th conference on ubiquitous computing*, UbiComp '11.
- [25] ZHANG, W., LI, S., AND PAN, G. Mining the semantics of origin-destination flows using taxi traces. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (2012), UbiComp '12.
- [26] ZHENG, Y., LIU, Y., YUAN, J., AND XIE, X. Urban computing with taxicabs. In *Proceedings of the 13th international conference on Ubiquitous computing* (2011), UbiComp '11.