This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3041263, IEEE Transactions on Big Data

SCCS: Smart Cloud Commuting System with Shared Autonomous Vehicles

Menghai Pan, Yanhua Li, Zhi-Li Zhang, Jun Luo

Abstract—
Emergence of autonomous vehicles (AVs) offers the potential to fundamentally transform the way how urban transport systems be designed and deployed, and alter the way we view private car ownership. In this paper we advocate a forward-looking, ambitious and disruptive *smart cloud commuting system* (SCCS) for future smart cities based on shared AVs. Employing giant pools of AVs of varying sizes, SCCS seeks to supplant and integrate various modes of transport – most of personal vehicles, low ridership public buses, and taxis used in today's private and public transport systems – in a unified, on-demand fashion, and provides passengers with a fast, convenient, and low cost transport service for their *daily commuting* needs. To explore feasibility and efficiency gains of the proposed SCCS, we model SCCS as a queueing system with passengers' trip demands (as jobs) being served by the AVs (as servers). Using a 1-year real trip dataset from Shenzhen China, we quantify (i) how design choices, such as the numbers of depots and AVs, affect the passenger waiting time and vehicle utilization; and (ii) how much efficiency gains (i.e., reducing the number of service vehicles, and improving the vehicle utilization) can be obtained by SCCS comparing to the current taxi system. Our results demonstrate that the proposed SCCS framework can serve the trip demands with 22% fewer vehicles and 37% more vehicle utilization, which shed lights on the design feasibility of future smart transportation systems.

Index Terms—Cloud Commuting, urban computing, queuing theory

1 Introduction

In most urban cities today, there are two primary modes of transit: i) *Public* transit services such as buses, subways which run along fixed routes with fixed timetables, and have limited coverage areas. These limitations mean that one cannot take public transport between any two arbitrary points in a city. ii) *private* transit services such as taxis, shared-van shuttles, (mobile app-based) ride-hailing services (e.g., Uber or Lyft) are largely "on-demand" – although their service may not be immediate or "real-time". However, taxi and ride-hailing services can be expensive, limiting them mostly for *ad hoc* use, namely, occasional short trips.

It is optimistic for us to imagine that, in the near future, there will be autonomous vehicles [1] (AVs) on the road networks either for commerce or private use. So far there are some exciting news about the application of AVs, for example, Voyage Auto, Optimus Ride and Waymo One have deployed robo-taxi systems in Florida, California and Arizona. Voyage Auto employs AV taxis to shuttle residents in a large retirement community in Florida. TuSimple, Kodiak, Ike Robotics, and Pronto. AI are developing long-haul autonomous driving system for trucks. Nuro.AI, Starship Technologies, Refraction AI and others are developing smaller, slower speed vehicle systems designed for last mile delivery (from a local warehouse to customer's home or business) of groceries and packages[22]. The emergence of autonomous vehicles although will offer new potentials to address the challenges facing the current urban transit systems, and challenge and transform how we view and design public and private transport systems in future smart cities. For instance, with their autonomy, would it still make sense to take "self-driving" cars to work, but have them spend most time parked, when in fact they can go somewhere by themselves? We envisage a forward-looking, ambitious and disruptive cloud commuting based transport system – *smart cloud commuting system* (SCCS) – for *future* smart cities based on *shared* AVs. Employing giant pools of AVs of varying sizes, SCCS seeks to supplant and integrate various modes of transport – most of personal vehicles, taxis, and low ridership public buses used in today's private and public transport systems – in a *unified*, *on-demand fashion*, and provides passengers with a *fast, convenient*, and *low cost* transport service for their *daily commuting* needs.

We postulate the four key aspects of *system efficiency gains* that could potentially be achieved in a smart cloud commuting system with shared AVs (see Section 2.1). This paper constitutes a first attempt at exploring the feasibility and efficiency gains of the proposed SCCS; due to space limitation, we focus primarily on the *temporal multiplexing gain through time-sharing of AVs*. To this end, we model SCCS as a queueing system with passengers' trip demands (as jobs) being served by the AVs (as servers). Using a 1-year real trip dataset from Shenzhen China, we quantify (i) how various design choices – such as the number of shared AVs and number and locations of *depots* (where idle AVs are stationed) – affect the passenger waiting time and vehicle utilization; and (ii) how much system efficiency gain (e.g., in terms of number of AVs and vehicle utilization) can be attained through SCCS.

- Utilizing a large-scale taxi trip dataset, we develop generative models to capture the arrival and service patterns of

- *Menghai Pan and Yanhua Li are with the Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA, 01609. E-mail: {mpan, yli15}@wpi.edu*
- *Zhi-Li Zhang is with the Department of Computer Science, University of Minnesota, Twin Cities, MN, USA E-mail: zhzhang@cs.umn.edu*
- *Jun Luo is with the Lenovo Machine Intelligence Center, Lenovo, HK E-mail: jluo1@lenovo.com*

1. Colloquially known as "self-driving cars" – however in our study we will use the term AVs to refer to not only passenger cars, but also "self-driving" shuttles, vans or busses; namely, AVs of *varying* sizes.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3041263, IEEE Transactions on Big Data

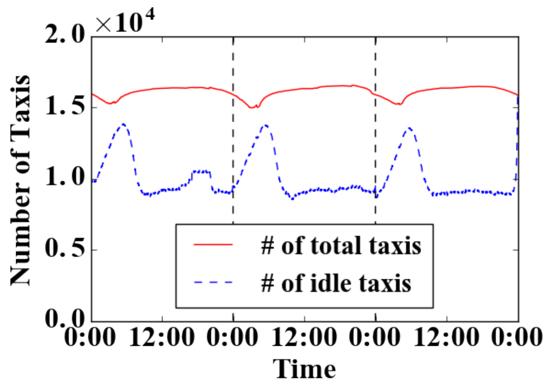


Fig. 1: Idling taxis

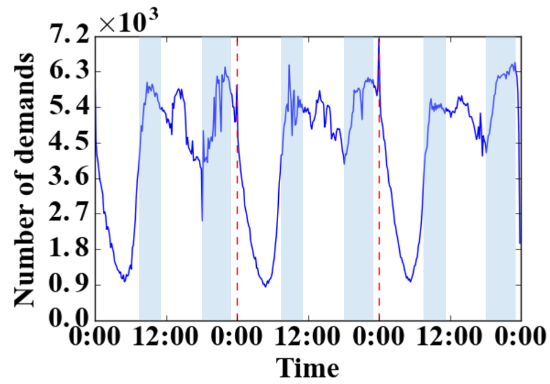


Fig. 2: Request pattern

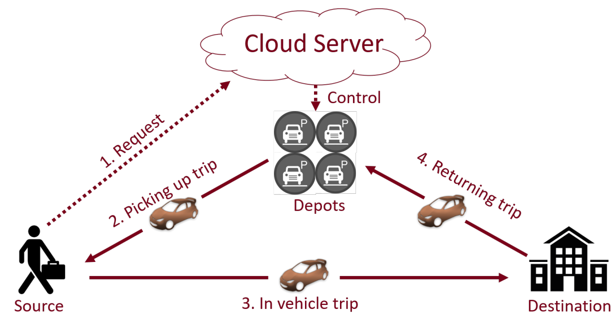


Fig. 3: Framework of SCCS

urban taxi trip demands over different time periods of the day.

- By modeling SCCS as an $M/G/k$ queuing system, we propose an theoretical framework to estimate the average waiting time of all passengers, given the total number of AVs and the number/locations of depots.
- We investigate the impacts of different design choices, e.g., number of AVs and number/locations of depots, on passenger waiting time and vehicle utilizations.
- We quantify the temporal multiplexing efficiency gain of time-sharing AVs achieved via SCCS, and compare that with the current urban taxi system. The evaluation results obtained using the 1-year taxi trip dataset demonstrate that the proposed SCCS can serve the trip demands with 22% less vehicles and 37% more vehicle utilization.

The rest of the paper is organized as follows. In Section 2, we motivate the proposed SCCS and outline a queueing system model for its feasibility study. In Section III we present the overall methodology and detail the modeling framework. In Section IV we describe the evaluation results using the Shenzhen taxi datasets. The related work is discussed in Section V, and the paper is concluded in Section VI.

2 Motivation and Problem Definition

In this section we first motivate the proposed SCCS. We then lay out a general queueing system model for SCCS for studying its feasibility and quantifying its potential efficiency gains.

2.1 Smart Cloud Commuting System (SCCS)

As alluded in the introduction, today's urban transit systems suffer many well-known shortcomings. Taking taxis as an example, Fig. 1 shows the number of on-road taxis in 3 days from $03/04/2014 - 03/06/2014$ for each 5-minutes time interval in Shenzhen, which indicates on average more than 60% of taxis are idle over time. Now imagine a (perhaps not-so-distant) future where we live in a smart city with autonomous vehicles or "self-driving" cars. How would the transport systems, both public and private, be designed in such a smart city? What transport services would be needed or plausible? Our envisaged SCCS is a bold attempt to re-imagine and re-design transport for future smart cities by fusing information technologies with AVs to offer a new kind of *mobility-as-a-service* that targets more specifically *daily commuting needs* for most (if not all) users in cities and metro areas (urban and suburban). As shown in Fig.3, in SCCS, each AV is controlled by (centralized) dispatch servers residing in the cloud. Once a passenger requests a trip, the cloud servers will arrange an AV to pick up and send the passenger to the destination. When a trip demand is completed, the vehicle can be re-used for other passengers. In this paper, we introduce the SCCS implemented with centralized servers, and in our future work, we will study the implementation of SCCS with decentralized cloud computing system. Our proposed SCCS can also benefit public transportation system with high-capacity AVs, e.g., autonomous buses. To integrate public transportation in SCCS, the high-capacity AVs can be assigned to pick up a group of passengers who can share the trips along the way. Employing giant pools of shared AVs of varying sizes, SCCS aims to provide users with a fast, convenient, and low cost transport service to meet their daily commuting needs. The *scale* and the resulting *abilities to maximize system efficiencies* via shared AVs differentiate our envisaged SCCS from today's ride-hailing services, which are designed primarily to serve *ad hoc* trips.

We postulate the following *four key aspects* of system efficiency gains that could potentially be achieved in a smart cloud commuting system with shared AVs. (i) *Temporal multiplexing gain through time-sharing of AVs:* by leveraging "bursty" travel demands and sharing of AVs over time, the number of AVs needed would be significantly less than what would be if every user had his or her personal AV. This is analogous to the statistical multiplexing gain attained by a packet-switched data network. (ii) *Payload multiplexing gain through ride-sharing among users*: By utilizing AVs of varying sizes to enable ride-sharing among users (similar to today's car-pooling, shared shuttle or transit services, but leveraging the autonomy of AVs), the number of AVs needed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3041263, IEEE Transactions on Big Data

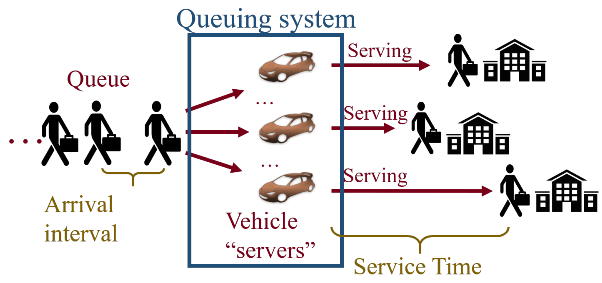


Fig. 4: Queuing system

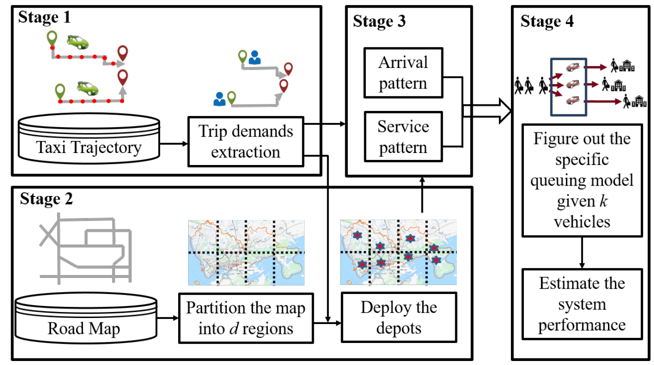


Fig. 5: Framework

can be further reduced. (iii) *Elastic demand gain through smart trip scheduling*: Many travel demands are elastic in nature (a trip to a store for grocery shopping now may not be crucial and thus can be delayed, say, for 30 minutes). Even for peak hour travel demands, as long as a user can reach her destination within a desired time window, the trip can be scheduled dynamically to leverage such elasticity to achieve additional system efficiency gain. (iv) *Road network efficiency gain through intelligent control of AVs:* With fewer vehicles on the road through shared AVs, road congestion can be alleviated or avoided, thus shortening trip times. Road network efficiency gain can be further increased by packing more AVs during peak demands (e.g., by reducing inter-car spacing) without creating safety issues, and by intelligent routing of AVs through less congested roads.

As a *first* attempt at studying the feasibility of the envisaged SCCS, in this paper we focus primarily on the first aspect of the system efficiencies, namely, *temporal multiplexing gain through time-sharing of AVs*, that can be potentially achieved through SCCS. In particular, by modeling SCCS as a queueing system, we investigate how various design choices – such as the numbers of vehicles and the number/locations of depots – affect the quality of services (QoS) of passengers (e.g., waiting time) and the overall system performance (e.g., vehicle utilization). For this study, we utilize a real-world, taxi trip dataset from Shenzhen, China over a period of one year. One interesting and important feature of this dataset lies in that due to the limited area coverage (and the fact that the public transit capacity cannot meet the demands during the *peak hours*), many residents in the city rely on taxis for daily commuting needs (see Fig. 2). This feature enables us to study the feasibility of the proposed SCCS to meet daily commuting needs and compare its system performance with that of the existing taxi system.

2.2 Modeling SCCS as a Queuing system

SCCS can be viewed as a queuing system. Passengers request for commute services from SCCS. Their requests will be placed in a queue, if the servers (i.e. AVs) are busy. Fig.4 shows the queuing model of SCCS, an arrival event is a request received from a passenger, and a service event is the process of an AV taking the passengers to the destination. As a queueing system, there are three components charactering the system performances, including the *arrival pattern*, *service pattern* and *number of servers*.

Arrival pattern is the distribution of the arrival events coming into the queuing system. We can use arrival rate and arrival interval to capture the arrival pattern of a queuing system. *Service pattern* captures the distribution of the service time.

Definition 1 (Arrival interval A). The arrival interval is the time period between each two successive trip requests.

Definition 2 (Arrival rate λ). The arrival rate is the number of trip requests arriving the system within a unit time slot.

Definition 3 (Service time S). The service time is the time period when a self-driving vehicle is dispatched to serve a passenger.

If the passengers' requests arrive the queue while all of the AVs are busy, the requests will be placed in a queue to wait for the next available AV. The waiting time indicates how long a passenger waits in a queue, which characterizes the quality of experience of the passenger in SCCS.

Definition 4 (Waiting time W). The waiting time is the time period from the arrival of a passenger request to an AV being dispatched to the passenger.

2.3 Problem Definition

Thanks to the fast development of location sensing technologies, the increasing prevalence of embedded sensors inside mobile devices, vehicles has led to an explosive increase of the scale of urban mobility datasets, including the trip demands data of passengers in urban areas.

Definition 5 (Trip demand). A trip demand of a passenger indicates the intent of a passenger to travel from a source location src to a destination location dst from a given starting time t_s with an expected trip duration Δt, which can be represented as a 4-tuple $\langle src, dst, t_s, \Delta t\rangle$.

Fig. 2 shows the temporal distribution of urban taxi trip demands for each 10-minute time interval in Shenzhen from $03/04/2014 - 03/06/2014$, which exhibits a clear diurnal pattern. Such pattern is driven by the daily commuting needs between residential and working locations. Given such strong diurnal pattern, we divide each day into a few time intervals, and focus on the daily dynamics of trip demands over intervals.

Problem definition. Given the total number of available self-driving vehicles k and the number of depots d, we aim to (1) estimate the impact of design choices (in k and d) on passenger waiting time and vehicle utilization; and (ii) evaluate the efficiency gains of SCCS comparing to the current taxi system, in terms of numbers of vehicles needed and the vehicle utilization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3041263, IEEE Transactions on Big Data

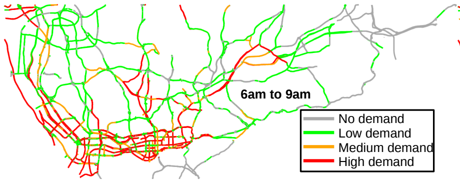


Fig. 6: Heat map of starting location

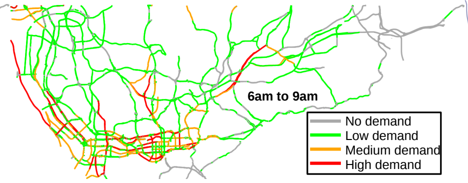


Fig. 7: Heat map of ending location

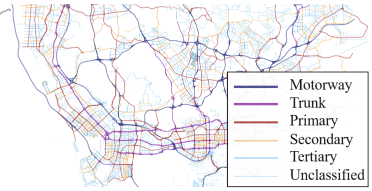


Fig. 8: Shenzhen road map

3 Methodology

In this section, we introduce our design model of SCCS given the total number of vehicles k and the number of depots d, and provides an analytical framework for analyzing the system performances and passenger quality of experience.

3.1 Overview

Fig. 5 illustrates our solution framework, that takes two sources of urban data as inputs and contains four key analytical stages: (1) trip demands extraction, (2) depots deployment, (3) arrival and service pattern extraction (4) system performance evaluation.

- **Stage 1 (Trip demands extraction)** This stage aims to extract the passengers' trip demands from the collected taxi GPS data. In our datasets, each taxi trajectory consists of a sequence of time-stamped GPS points, where a GPS point is collected every 40 seconds on average. A GPS data point includes the time stamp, latitude, longitude, and binary indicator (indicating if a passenger is aboard). Moreover, the raw trajectory data are noisy, with spatial errors from the groud-truth locations, due to the accuracy limit of the GPS devices. By cleaning the taxi GPS data, we can extract the passenger taxi trips, indicated by four key elements: (1) starting location src, (2) ending location dst, (3) starting time t_s, (4) trip duration Δt. As a result, each trip represents a passenger demand.
- **Stage 2 (Depots deployment)** Given the number of depots d and the number of AVs k, this stage aims to identify the depot locations and assign AVs to depots. First, the urban area is divided into d grids with equal sizes. Second, the trip demands extracted in stage 1 can be aggregated into each grid based on the source locations. Then, for each grid with trip demands, we will deploy a AV depot. To reduce the dispatching distance, the depot location is obtained by the average geo-location of all trip source locations inside the grid. If the location is not exactly on a road segment, the depot location will be shifted to the nearest road network.
- **Stage 3 (Arrival/Service pattern extraction)** With a particular SCCS design (from stage 2), this stage will examine the arrival and service patterns. The trip requests arrive in a sequence of time stamps, i.e., $\{t_{s_1}, t_{s_2}, ..., t_{s_m}\}$. We will quantify the arrival pattern of such time sequence. Moreover, with all trip durations (as system service times), we will characterize the service pattern.
- **Stage 4 (System performance estimation)** With generative models for arrival and service patterns of the urban trip demands, we can naturally view the taxi service system as a queuing system, with trip demands as the customers and taxis as the servers. In Stage 4, by modeling the SCCS as an $M/G/k$ queueing system, we will quantify the average waiting time of passengers and vehicle utilizations.

Fig. 9: GPS set and passenger indicator on taxis

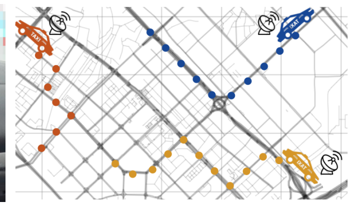


Fig. 10: GPS data generated by taxis

3.2 Data Description

Our analytical framework takes two urban data sources as input, including (1) taxi trajectory data and (2) road map data. For consistency, both datasets are collected in Shenzhen, China in 2014. We introduce the details of these datasets below.

Taxi trajectory data are GPS records collected from taxis in Shenzhen, China during 2014. There were in total $17,877$ taxis equipped equipped with GPS sets and passenger indicators as shown in Fig.9, where each GPS set generates a GPS point every 40 seconds on average. The passenger indicator will be pulled down if there is a passenger aboard, and it sends binary values indicating if a passenger is aboard or not to the GPS set. Overall, a total of $51,485,760$ GPS records are collected on each day, and each record contains five key data fields, including taxi ID, time stamp, passenger indicator, latitude and longitude. The passenger indicator field is a binary value, indicating if a passenger is aboard or not. Note that, in this paper, the results are from the data collected in 2014, and similar results can be obtained with new data collected in 2016. For better reproducibility, we made our data public.[2]

TABLE 1: Road Map Data in Shenzhen

Type	Counts	Type	Counts
Motorway	563	Secondary	868
Trunk	258	Tertiary	1,393
Primary	745	Unclassified	16,829

Road map data. In our study, we use Google GeoCoding [1] to retrieve a bounding box of Shenzhen, which is defined between $22.44°$ to $22.87°$ in latitude and $113.75°$ to $114.63°$ in longitude. The covered area covers a total of $1,300km^2$. Within such a bounding region, we crawl road map data in Shenzhen from OpenStreetMap [3]. The road map data contain six levels of road segments, which are detailed in Table 1 and visualized with different colors in Fig.8.

2. https://www.dropbox.com/sh/m28gv5qeh7wbeft/AAAj92KoycrhSdX6Q4hBu4lMa?dl=0

3.3 Stage 1: Demands Extraction

In stage 1, we clean and extract the urban trip demands from the raw trajectory data.

Trajectory data cleaning. The trajectory data are noisy in nature. First of all, the GPS locations are with errors of around 15 meters. Secondly, there are GPS points outside the bounding box of Shenzhen. We conduct two steps to clean the noisy trajectory data, including map-matching and spatial filtering. *Map-matching* is a process that project the noisy GPS locations back to the road segments, which has been extensively studied in the literature We apply the map-matching technique [11] to our dataset. Secondly, we apply a simple *spatial filtering* step to remove GPS records that are outside the bounding region of Shenzhen.

Trip demand extraction. The passenger indicator field in the taxi trajectory data is the key enabler to extract the taxi trip demands. A taxi trip can be represented as a sequence of taxi GPS points with the passenger indicator as 1. The first and last GPS locations of the taxi trip capture the source/destination locations (src, dst) of a trip demand, and the corresponding time stamps characterize the trip starting/ending time t_s/t_e. The trip duration can be obtained as the elapsed time from t_s to t_d, i.e., $\Delta t = t_e - t_s$. Once we have all trip demand tuples $\langle src, dst, t_s, \Delta t\rangle$, we observe that there are a small number of trip demands with extremely short or long trip durations. From the size of the bounding region of Shenzhen and the road map, any trip could be done within 2 hours (including the rush hours with traffic congestion). Moreover, people would not take a taxi trip shorter than 2 minutes in general. Thus, we simply filter out those noisy taxi trips longer than 2 hours or shorter than 2 minutes, which may be due to the issues with hardware or data collection processes. Note that in this work, each demand can be from either one individual passenger or a group of passengers sharing the entire trip. Without loss of generality, we assume each demand is from one passenger.

After the two steps, we obtain a total of $595,501$ daily trip demands from our trajectory data. Fig.6 and Fig.7 show the geo-distributions of source and destination locations in Shenzhen during the morning rush hours 6–9AM on March 6th, 2014.

3.4 Stage 2: Depots deployment

Given the number of depots d and total number of available vehicles k, our system deployment model works as follows: (1) road map partitioning, (2) depot placement, (3) vehicles assignment.

Step 1: Road map partitioning. We first get the boundary of Shenzhen from OpenStreetMap, which is defined between $22.44°$ to $22.87°$ in latitude and $113.75°$ to $114.63°$ in longitude. Then, we partition the area of the city into d grids with the sizes.

Step 2: Depot placement. After the regions are divided, we try to deploy one depot in each region, and totally d depots will be deployed. First, we aggregate the trip demands extracted in stage 1 into each grid. In SCCS, the request in a grid will be served by the depot in that region. We allocate those demands into grids based on their source locations. Then, to reduce the dispatching distances, in each grid, the center location of all the source demand locations are calculated to place the depot. Moreover, if the center source locations is not on the road network, it will be shifted to the nearest road segment. Fig.11 shows the result of road map partition and depot deployment. Note that one region is in the ocean, and we do not deploy a depot in that region.

Step 3: Vehicle assignment. After deploying the depots, the vehicles are assigned to each depot according to the portion of demands in the region. Let N be the total demands in the urban area, N_i be the number of demands in region i. The total number of vehicles assigned to region i is thus $k_i = k \cdot N_i/N$.

TABLE 2: Parameters of arrival rate distributions

Time slot	12am-6am	6am-12pm	12pm-6pm	6pm-12am
λ	4.1375	7.6189	8.4415	9.0023

3.5 Stage 3: Arrival/Service pattern

SCCS can be viewed as a *queuing system*. Each trip demand and the corresponding trip represent a customer arrival event and a service event, respectively. Self-driving vehicles are the servers in the system. Now we characterize the arrival pattern and service pattern from the trips.

Arrival pattern analysis. We chose the time unit as one second, and count the number of arrived trip demands over each second in demand data we obtained from Stage 1. Fig.12 shows the distributions of the arrival rate in four different intervals of a day. The x-axis represents the arriving rates and the y-axis is the percentage of demands. The blue dots are obtained from original demands data, which nicely fit Poisson distributions. The green curves are the best fitting curves with Poisson distribution. The parameters λ's of Poisson distributions are the mean arrival rates, which are listed in Table 2 for different time intervals in a day.

Service pattern analysis. As shown in Fig.3, the service time of an AV include three time intervals. The first part is *pickup time*, namely, the passenger sends a request to the cloud servers to request a trip service. The cloud servers arrange a vehicle to pick the passenger up, if there is an available vehicle in the depot, otherwise, the passenger would wait in the queue. After the vehicle picked up the passenger, it will take the customer to the destination, during which the passenger experiences *in-vehicle time*. When the trip is completed, the vehicle returns to the nearest depot to the passenger dropoff location, which is the *return time*.

Note that a complete service time include all three time intervals, i.e., pickup, in-vehicle, and return times. Though passenger does not experience the return time, it is counted, because the vehicle is still "reserved" and cannot serve other passengers (on the trip back to the depot)[3].

Since each request will be served by a vehicle from the depot in the source region, and the destination of the demand may be in a different region, a vehicle balancing approach is required. We adopt a simple schedule-based approach for vehicle rebalancing: Every 12 hours, the vehicles will be rebalanced to the initial numbers of vehicles. Moreover, the on-road travel time can be estimated by OSRM API [2] from one place to another. Thus, the picking up time and the returning time of each demand can be estimated by the API.

To extract the service time pattern from the demand data, we choose the unit time as minute. Taking $k = 12000$ as an example, Fig.13 show the distributions of service time given different number of depots: 1,2,3,4,8,16 depots, in the 12pm-6pm time slot on March 5th in 2014. The x-axis represents the service time and the y-axis is the percentage of demands. The black dots are from the raw demand data, which cannot be fitted by

3. Note that the system can be further designed to allow vehicles to direct pick up the next passengers without going back to depot, which require more complex system design model. To simplify our feasibility and performance gain analysis, we adopt this simple model, and leave it for our future work to evaluate more complex system design.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3041263, IEEE Transactions on Big Data

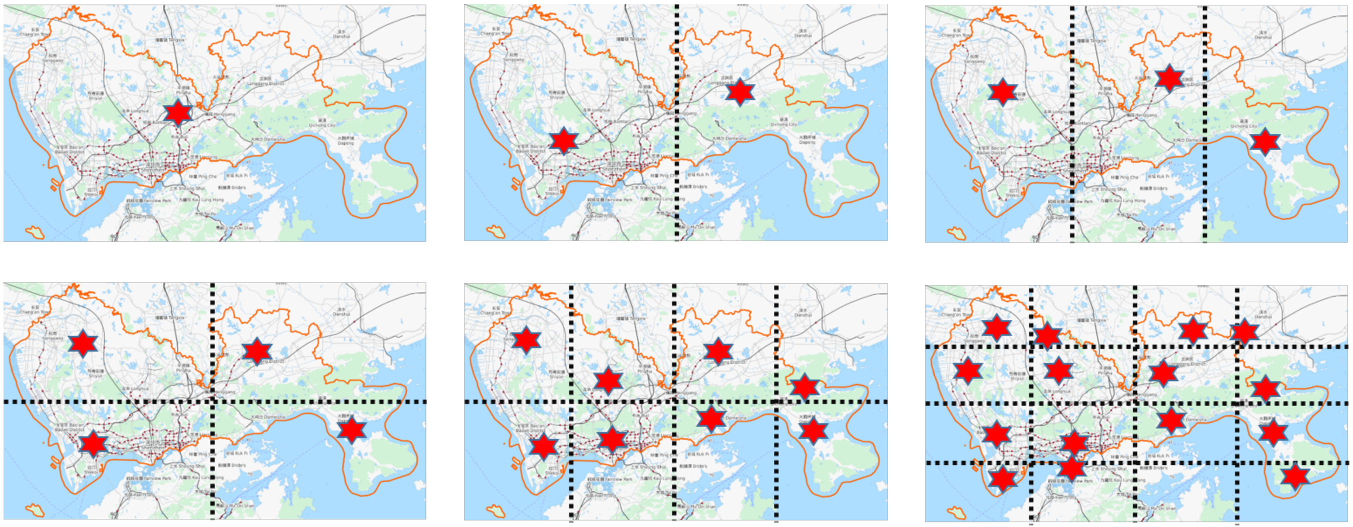
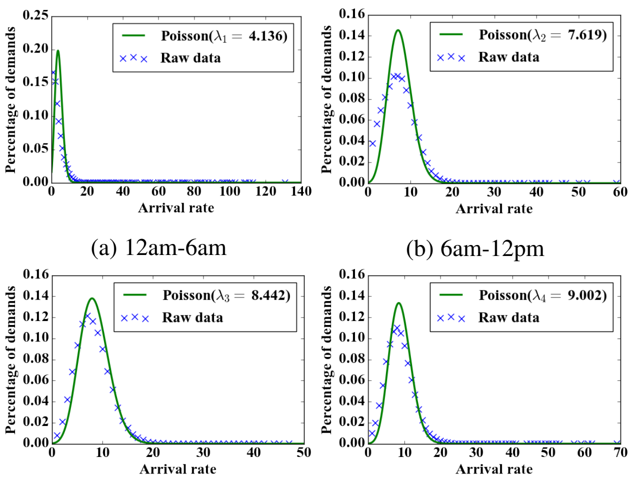


Fig. 11: Depot placement in Shenzhen



(a) 12am-6am

(b) 6am-12pm

(c) 12pm-6pm

(d) 6pm-12am

Fig. 12: Arrival rate (#requests/s)

TABLE 3: Average service time

# Depots	1	2	3	4	8	16
$\overline{S}$(min)	58.45	51.67	49.80	41.31	31.60	29.38

a simple distribution. Hence, the service pattern follows a general distribution, denoted as G in queueing theory. The average service times with different number of depots are listed in Table 3.

3.6 Stage 4: Estimating the system performance

Now, we are in a position to introduce our queuing theory based approach to estimate the average waiting time in SCCS, given the number of available vehicles k.

We have shown that the trip demands arrival rate follows a Poisson distribution, but the service pattern is general. When k vehicles are available in SCCS, we can denote this queuing system as an $M/G/k$ queue. It is still an open question to exactly quantify the features of such a queue, such as waiting time [13]. We employ the approximation algorithm [15] to estimate the average waiting time in $M/G/k$ queue by adjusting the mean waiting time in a corresponding $M/M/k$ queue. Equation (1) shows the approximation function of the average waiting time in $M/G/k$ queue. where $E[W^{M/G/k}]$ and $E[W^{M/M/k}]$ are the expected waiting times of the $M/G/k$ and $M/M/k$ queues, respectively. The $M/M/k$ queue has the same mean service time as the $M/G/k$ queue.

$$E[W^{M/G/k}] = \frac{C^2+1}{2} E[W^{M/M/k}] \tag{1}$$

where C is the coefficient of variation of the service time distribution in $M/G/k$ queue. In $M/M/k$ queue, the average waiting time can be calculated in Eq (2).

$$E[W^{M/M/k}] = \frac{Er_c(k,\rho)\overline{S}}{k-\rho}, k > \rho \tag{2}$$

where ρ is the utilization in a queuing system, which equals to $\lambda\overline{S}$, and $Er_c(k,\rho)$ is the Erlang C formula(Eq (3)), which indicates the probability that an arriving customer has to wait, which is also the proportion of time that all k servers are busy. $k > \rho$ ensures the system can reach the steady state.

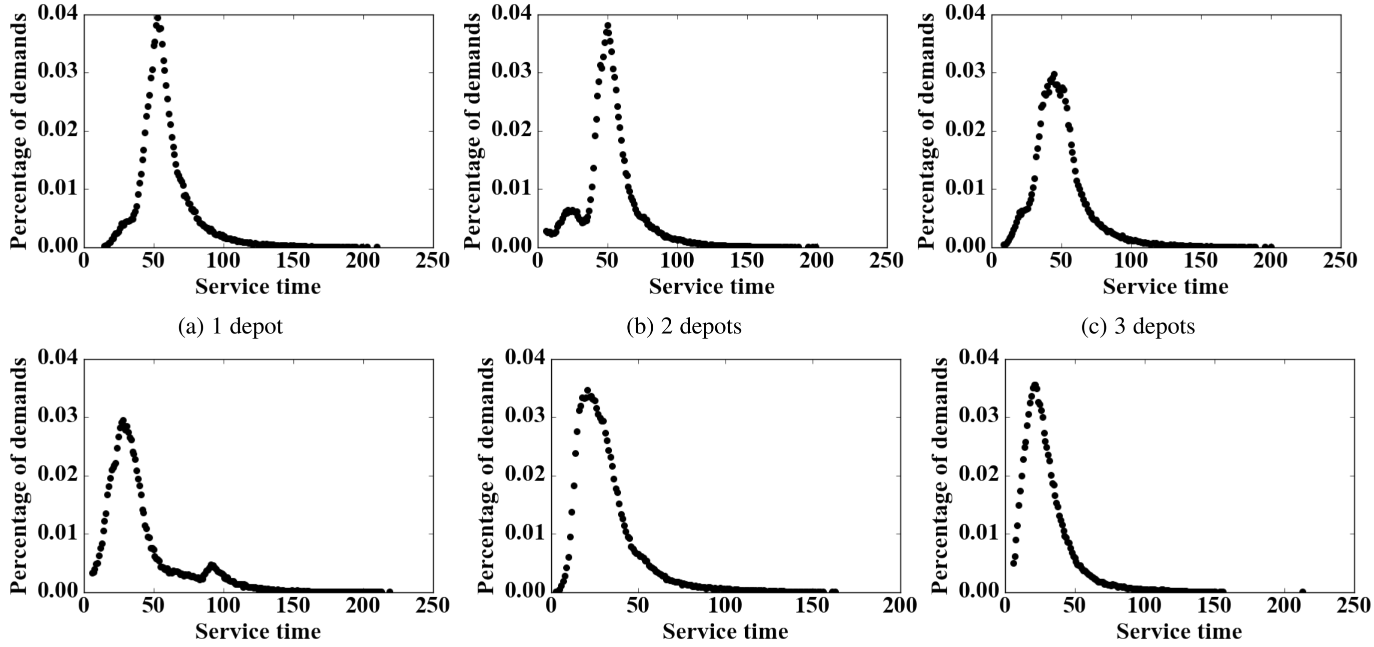
$$Er_c(k,\rho) = \frac{\frac{k\rho^k}{(k-\rho)k!}}{\sum_{k=0}^{k-1}\frac{\rho^n}{n!} + \frac{k\rho^k}{(k-\rho)k!}} \tag{3}$$

Finally, we can approximate the average waiting time in $M/G/k$ queue. Taking one depot deployment as an example, the arrival rate in $12pm - 6pm$ slot is 5.0594, and the average service time of the system is 3536.45249, so the utilization $\rho = 17876.4137$, and the coefficient of variation of the service time distribution $C = 0.5563$. Given the number of vehicles $k = 18000$, we can first get $Er_c(18000, 17876) = 0.2547$, which means that 25.47% of the time when all of the servers are busy. Finally the approximate average waiting time is 4.0134 seconds.

4 Evaluation

In this section, we use real taxi trip data to conduct experiments to evaluate (1) the performance of the design choices of number of available vehicles k and the number depots d. (2) the efficiency gain in SCCS comparing with current taxi system.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2020.3041263, IEEE Transactions on Big Data



(a) 1 depot (b) 2 depots (c) 3 depots

(d) 4 depots (e) 8 depots (f) 16 depots

Fig. 13: Service time($k = 12000$)

4.1 Evaluation settings

Time intervals in a day. We observe that the trip demand arrival and service patterns change dramatically over time intervals in a day. In our evaluations, we divide a day into 4 time intervals, we have the cutting-off times as $[12am, 6am, 12pm, 6pm]$. and evaluate how the granularities affect the performances of our proposes models.

Baselines. We compare the performances of our SCCS (in different design choices) with the current taxi system. To evaluate how our SCCS performs when serving the same set of trip demands in our taxi data, we employ a data-driven simulation approach as follows: The real world trip demands arrive by the order of their starting times. If there are available vehicles in its regional depot, the waiting time of this demand will be 0. Otherwise, the waiting time is the time interval from the starting time to the moment when a vehicle returns to that depot. The results introduced below show that our SCCS can achieve several efficiency gains comparing with the current transit system in vehicle utilization and number of vehicles needed.

Metrics. For the design choices, we use the customer in system time and vehicle idle rate to evaluate the performance of the system. The efficiency gain is evaluated by the number of vehicles needed, and the utilization of the vehicles while serving the same amount of demands in our system and current urban taxi transit system.

4.2 Design choices

4.2.1 Impact of k

From the passengers' perspectives, the service process consists of two parts: passenger waiting time and in-vehicle time. The passenger waiting time includes the system waiting time W[4] (as defined in Sec 2-B) and the picking up time. We denote the total service time passenger experienced as the in-system time, namely, the total of waiting time, pickup time, and in-vehicle time. The in-system time is what passenger actually experiences, and is considered as the quality of service the passenger received.

Taking 16 depots as an example, given the number of vehicles $9000, 10000, 11000, 12000, 15000, 20000$, we can simulate the whole service in our SCCS, and get the passenger in-system time, which is shown in Fig.14. We can observe that as we increase the number of vehicles, the passenger in-system time decreases.

Moreover, Fig. 20 shows the average in-system time and the idle rate for different numbers of AVs. With the increase of the total number of vehicles, the in-system time decreases, which is because the waiting time becomes shorter. However, the idle rate, which characterizes the portion of time that a vehicle stays idle in the depot (Eq (4)), increases due to the increasing number of over-deployed AVs.

$$R_{idle} = \frac{\sum_{i=1}^{k} T_{idle}^{i}}{k \cdot T}, \tag{4}$$

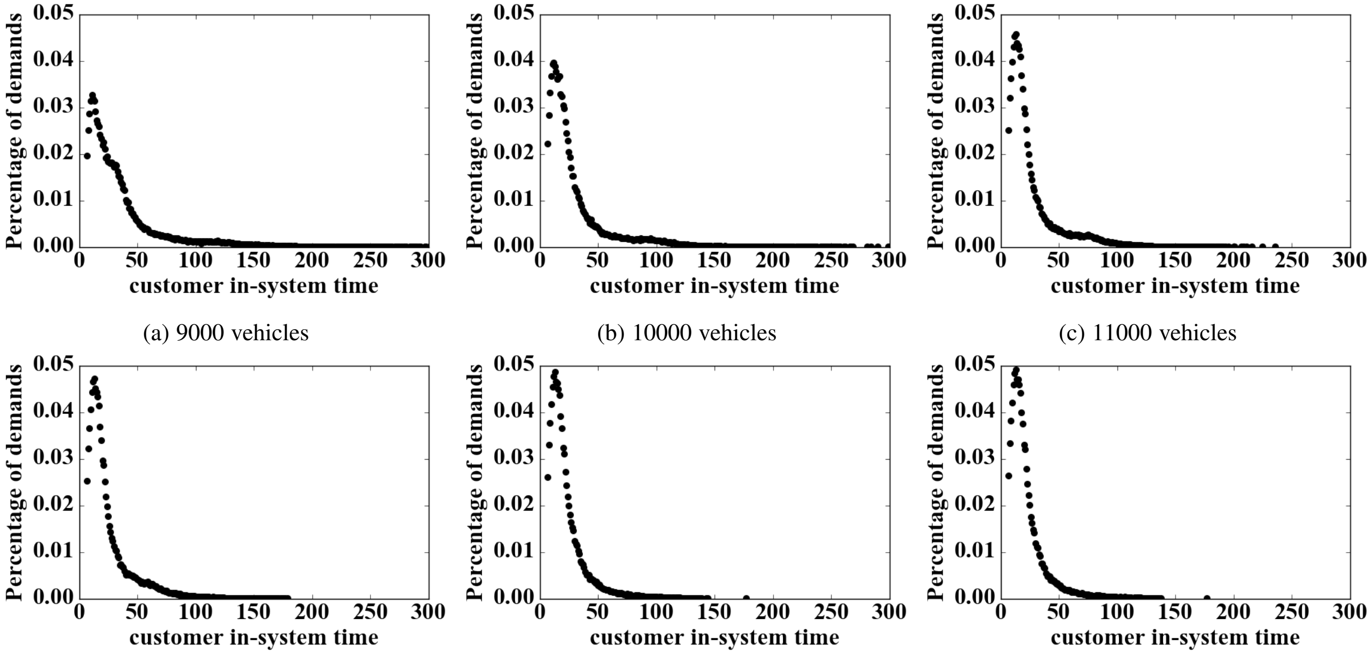
with T as the total amount of time in a day (i.e., 24 hours), and T_{idle}^{i} is the amount of time the vehicle i spent in depot during the day.

Fig. 20 clearly indicates the trade-off between the waiting time and the idle rate when changing the number of vehicles.

The number of depots in our system can also have effects on the customer's experience. Taking $k = 12000$ for example, Fig. 15 shows the change of the customer in-system time according to the number of depots, when we fixed the number of AVs to be 12000. Fig. 15(a)–(f) shows that as we increase the number of depots, the passenger in-system time distribution evolutes from high to low in-system time. Moreover, Fig. 16–17 indicates how the average in-system, waiting time changes, over different numbers of depots.

The phenomena occur because the increase of the number of depots can reduce the picking up time and the waiting time for each service. Moreover, from Fig. 15, we can clearly observe that

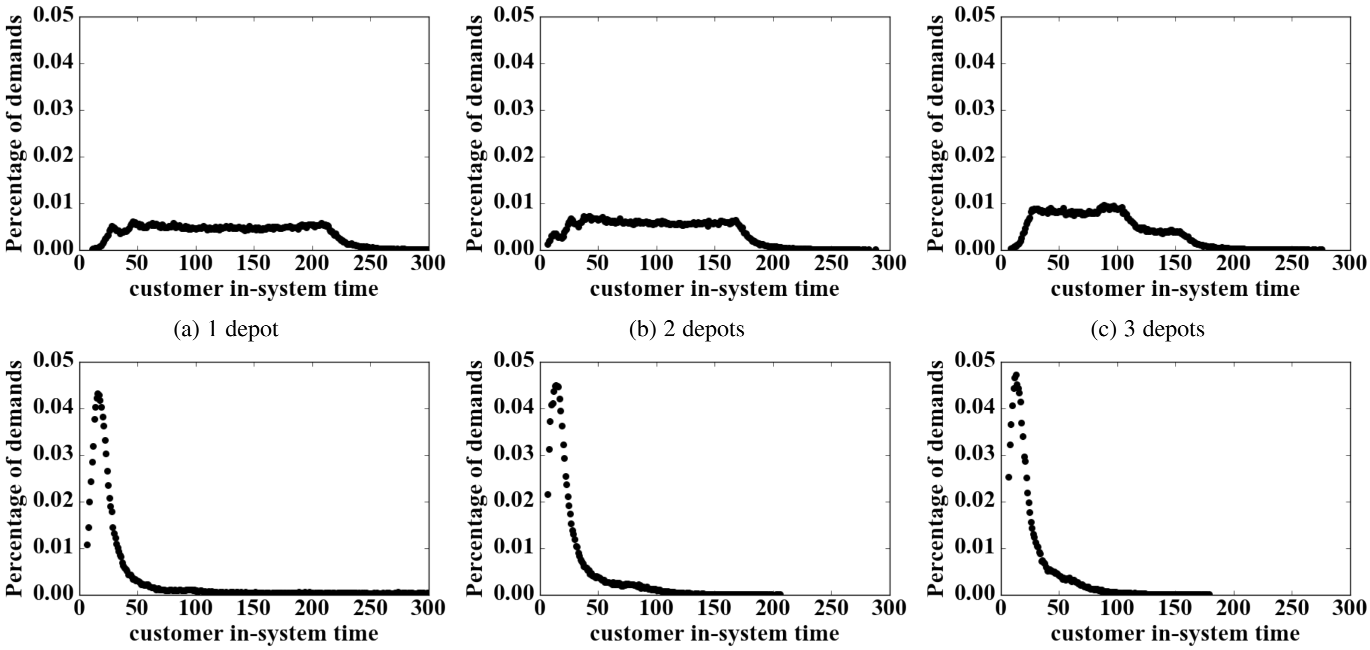
4. Note that the system waiting time is different from the passenger waiting time, where the former is the time from the request arrival to the time a vehicle is dispatched, and the latter includes both the system waiting time and pickup time.



(a) 9000 vehicles (b) 10000 vehicles (c) 11000 vehicles

(d) 12000 vehicles (e) 15000 vehicles (f) 20000 vehicles

Fig. 14: the impact of total number of taxis



(a) 1 depot (b) 2 depots (c) 3 depots

(d) 4 depots (e) 8 depots (f) 16 depots

Fig. 15: The impact of number of depots (k=12,000)

4.3 System efficiency gains

By comparing our SCCS with the current taxi system, we now show that the SCCS system can achieve efficiency gains in several aspects, including (1) the higher vehicle utilization, (2)the less number of vehicles needed. Here, the results presented are from the real-world data collected in Shenzhen, China, 2014, the traffic volume does not show significant difference over time, and similar results of vehicle utilization and number of vehicle needed can be obtained with data collected in different time (2016).

4.3.1 Utilization of vehicles

In Fig. 1, we show that most of the taxis are idling on the road over days, which means the utilization of the taxis in current taxi system is low. At each time slot, e.g., in 1 hour, we can obtain a ratio of in-service vehicle vs the total number of vehicles. We quantify the utilization of the vehicles as average ratio of in-serve vehicles over all time slots, defined as follows.

$$U = \frac{\sum_{i=1}^{T_{slots}} (N_i^{busy}/N_i)}{T_{slots}}, \tag{5}$$

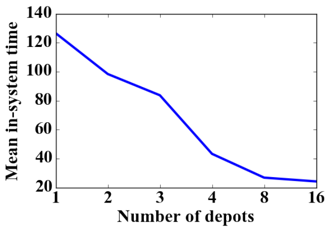


Fig. 16: Average in-system time

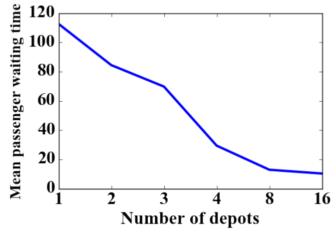


Fig. 17: Average passenger waiting time

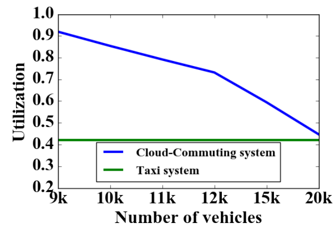


Fig. 18: utilization of vehicles

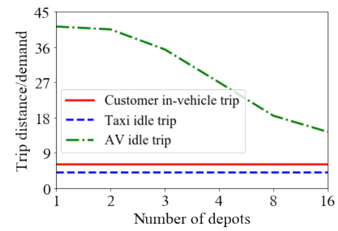


Fig. 19: Distance per demand in SCCS and taxi system

TABLE 4: V-values

# Vehicles	7k	7.5k	8k	9k	12k	20k
V	0.465	0.426	0.429	0.462	0.586	0.801

where T_{slots} is the total number of time slots in a day, N_i^{busy} and N_i are the number of in-service and all vehicles at time slot i.

The utilization of the vehicles in our system is shown in Fig. 18 when $d = 16$. Taking $k = 11000$ as an example, the utilization is 79.1%, while the utilization of the taxis in Shenzhen was 42.02%.

4.3.2 Number of vehicles needed

We can count the number of taxis in Shenzhen taxi system from our trajectory data, which was in total $9,606$ taxis. When using SCCS to serve the same trip demands, the number of vehicles would have impacts on the trade-off between the passenger in-system time and the vehicle idle rate (see Fig. 20). We define a measure V-value in Eq.(6) as a combination of the two measures to quantify the system performance.

$$V_k = \alpha T_{in-system} + (1-\alpha) \cdot R_{idle}, \tag{6}$$

with α as a design trade-off parameter within $[0, 1]$. The smaller V-value indicates better performance. Taking 32 depots and $\alpha = 0.01$ as example, the V-values are listed in Table 4. The most appropriate number of vehicles in 32 depots is 7500, which shows a 22% reduction on needed vehicles.

4.4 Travel Distance Analysis

In this section, we compare the travel distances of AVs in SCCS with those of taxis in current taxi system. Fig.19 shows the average distance per demand for the customer in-vehicle trips and idle trips of AVs in SCCS and taxis in current taxi system. As shown in Fig.3, the **AV idle trip** includes the picking up trip and the returning trip, and the distances of these trips are obtained via the OSMnx API[6]. The **taxi idle trip** is the trip when the taxi has no passengers onboard. The distance of **customer in-vehicle trip** is the same in SCCS and taxi system. The distances of the later two types of trips are extracted from the taxi GPS data. From Fig.19, we find that, as the number of depots increase in SCCS, the average AV idle trip distance per demand decreases, because the distances of picking up and returning trips decreases when there are more depots over the city. Although the average AV idle trip distance is higher than that of taxis in current taxi system when there are 16 depots in the city, it is safe for us to estimate that when the number of depots is sufficiently large, the average idle distance of AVs in SCCS will be lower than that of taxis in current taxi system.

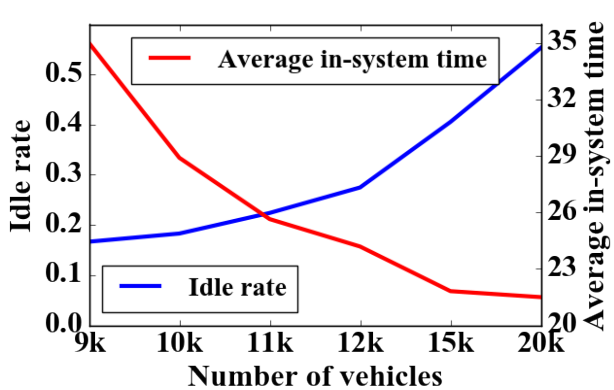


Fig. 20: Tradeoff

5 Related Work

To the best of our knowledge, we are the first to propose a Smart Cloud Commuting System (SCCS) for future smart cities with AVs, and quantify its feasibility and efficiency gains. In this section, we introduce two research areas that are related to our work, including (1) mobility-on-demand system, and (2) urban computing.

Mobility-on-demand system (MoD). MoD ([21], [4], [29], [23], [26], [31], [20]) is an emerging concept in solving urban transportation problems, such as unbalanced supply-demand rates and traffic congestion. MoD aims to provide transit supplies, such as shuttle/taxi services according to dynamic urban trip demands. In [21], authors design a simulation platform to explore the performance of autonomous vehicle based MoD system under various vehicle dispatching models. In another work [4], a general mathematical model is proposed, which could make real-time assignment decision in high-capacity ride-sharing system. This model is designed to handle a large number of passenger demands and dynamically generate optimal assignment solution to urban trip demands. In [29] and [23], authors propose two spatial queueing-theoretical models, that capture salient dynamic and stochastic features of customer demand, for Autonomous mobility-on-demand system which has autonomous vehicles in it. In [12], [5], [7], [8], the authors envisioned mobility systems with AVs, and analyzed the performance of the system via simulation. [24] provides a review of recent studies on investigating the impacts of AVs on travel behaviour and land use. Differing from these works with focus on the (ride-sharing) dispatching algorithms for load balancing of vehicles, we employ real world data (rather than simulation) to analyze the underlying trip demand patterns with queuing theory and evaluate design trade-offs and efficiency gains under a unifying SCCS framework.

Urban Computing is a thriving research area which integrates urban sensing, data management and data analytic together as a unified process to explore, analyze and solve crucial problems related to people's everyday life [16], [19], [25], [10], [17], [28], [18], [9], [27], [30], [14]. For examples, [16] presents a data-driven optimization framework to deploy charging stations and charging points with the goal of minimizing the seeking and waiting time of electric vehicle drivers. [25] develops novel models to predict future crowd flow traffic in subway stations. [27] introduces a method to estimate the travel time in a road segment using sparse trajectories data. [28] proposes a model to discover urban function zones by exploring latent activity trajectory data. In [30], the authors propose a method to diagnose the noises environment in New York city by extracting ubiquitous data over the city. Differing from these works, in this paper, we propose a future smart cloud commuting system (SCCS) with shared autonomous vehicles, and quantitatively evaluate the feasibility and efficiency gains of SCCS.

6 Conclusion and Future Work

In this paper, we advocate a Smart Cloud Commuting System (SCCS) for future smart cities with shared AVs to meet daily commuting demands of a large urban city. We have outlined four aspects of system efficiencies that can potentially be attained via the envisaged SCCS. As a first attempt at studying its feasibility, in this paper we develop generative models to capture fundamental trip demand arrival and service patterns, and develop a novel framework to explore the impact of design choices on the temporal multiplexing gains (through time-sharing of AVs) that can be achieved by SCCS. We conducted extensive evaluations using a large scale urban taxi trajectory dataset from Shenzhen, China. The results demonstrate that SCCS can reduce the number of vehicles by 22%, and improve the vehicle utilization by 37%.

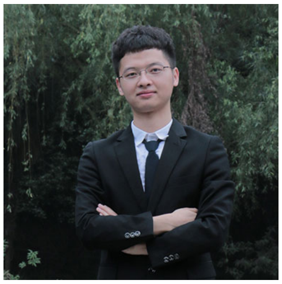
As part of our future work, we plan to further incorporate the vehicle rebalancing algorithms that allow vehicles to serve other passengers without going back to depots in this study. Furthermore, we will extend our current modeling framework to investigate the other three aspects of the system efficiencies afforded by the envisaged SCCS by the effects of ride-sharing, smart trip scheduling and AV routing, and so forth.

7 Acknowledgement

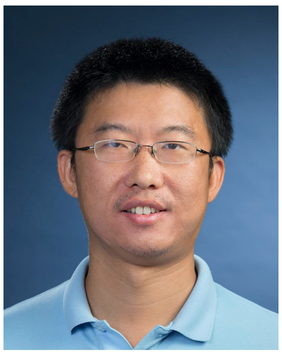
Menghai Pan and Yanhua Li were supported in part by NSF grants IIS-1942680 (CAREER), CNS-1952085, CMMI-1831140, and DGE-2021871.

References

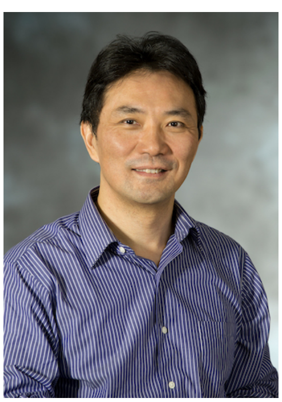
[1] Google GeoCoding. https://developers.google.com/maps/documentation/geocoding/.
[2] Open Source Routing Machine. https://http://project-osrm.org/.
[3] OpenStreetMap. http://www.openstreetmap.org/.
[4] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 2017.
[5] J. Bischoff and M. Maciejewski. Simulation of city-wide replacement of private cars with autonomous taxis in berlin. *Procedia Computer Science*, 83, 2016.
[6] G. Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.
[7] P. M. Boesch, F. Ciari, and K. W. Axhausen. Autonomous vehicle fleet sizes required to serve different levels of demand. *Transportation Research Record*, 2542(1):111–119, 2016.
[8] T. D. Chen, K. M. Kockelman, and J. P. Hanna. Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions. *Transportation Research Part A: Policy and Practice*, 94:243–254, 2016.
[9] Y. Ding, Y. Li, K. Deng, H. Tan, M. Yuan, and L. M. Ni. Dissecting regional weather-traffic sensitivity throughout a city. In *ICDM*, 2015.
[10] Y. Ding, Y. Li, K. Deng, H. Tan, M. Yuan, and L. M. Ni. Detecting and analyzing urban regions with high impact of weather change on transport. *IEEE Transactions on Big Data*, 2016.
[11] H.-P. Hsieh, S.-D. Lin, and Y. Zheng. Inferring air quality for station location recommendation based on urban big data. In *SIGKDD*, 2015.
[12] W. C. Jordan. Transforming personal mobility. 2012.
[13] J. Kingman. The first erlang century—and the next. *Queueing systems*, 63(1-4):3, 2009.
[14] C. Kumar, D. Basu, and B. Maitra. Modeling generalized cost of travel for rural bus users: a case study. *Journal of Public Transportation*, 2004.
[15] A. Lee and P. Longton. Queueing processes associated with airline passenger check-in. *OR*, pages 56–71, 1959.
[16] Y. Li, J. Luo, C.-Y. Chow, K.-L. Chan, Y. Ding, and F. Zhang. Growing the charging station network for electric vehicles with trajectory data analytics. In *ICDE*, 2015.
[17] C. Liu, K. Deng, C. Li, J. Li, Y. Li, and J. Luo. The optimal distribution of electric-vehicle chargers across a city. In *ICDM*. IEEE, 2016.
[18] X. Liu, X. Kong, and Y. Li. Collective traffic prediction with partially observed traffic history using location-base social media. In *CIKM*, 2016.
[19] B. Lyu, S. Li, Y. Li, J. Fu, A. C. Trapp, H. Xie, and Y. Liao. Scalable user assignment in power grids: a data driven approach. In *SIGSPATIAL GIS*. ACM, 2016.
[20] M. Maciejewski, J. Bischoff, and K. Nagel. An assignment-based approach to efficient real-time city-scale taxi dispatching. *IEEE Intelligent Systems*, 2016.
[21] K. A. Marczuk, H. S. S. Hong, C. M. L. Azevedo, M. Adnan, S. D. Pendleton, E. Frazzoli, et al. Autonomous mobility on demand in simmobility: Case study of the central business district in singapore. In *CIS-RAM 2015*.
[22] T. Nandwana. Autonomous vehicles: Why drive when the vehicle drives you.
[23] M. Pavone. Autonomous mobility-on-demand systems for future urban mobility. In *Autonomous Driving*, pages 387–404. Springer, 2016.
[24] A. Soteropoulos, M. Berger, and F. Ciari. Impacts of automated vehicles on travel behaviour and land use: an international review of modelling studies. *Transport reviews*, 39(1):29–49, 2019.
[25] E. Toto, E. A. Rundensteiner, Y. Li, R. Jordan, M. Ishutkina, K. Claypool, J. Luo, and F. Zhang. Pulse: A real time system for crowd flow prediction at metropolitan subway stations. In *ECML-PKDD*. Springer, 2016.
[26] R. Wang, C.-Y. Chow, Y. Lyu, V. C. S. Lee, S. Kwong, Y. Li, and J. Zeng. Taxirec: Recommending road clusters to taxi drivers using ranking-based extreme learning machines. In *SIGSPATIAL*, 2015.
[27] Y. Wang, Y. Zheng, and Y. Xue. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.
[28] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 2015.
[29] R. Zhang, K. Spieser, E. Frazzoli, and M. Pavone. Models, algorithms, and evaluation for autonomous mobility-on-demand systems. In *American Control Conference (ACC), 2015*, pages 2573–2587. IEEE, 2015.
[30] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, and E. Chang. Diagnosing New York City's Noises with Ubiquitous Data. In *Ubicomp*, 2014.
[31] X. Zhou. Dynamic origin-destination demand estimation and prediction for off-line and on-line dynamic traffic assignment operation. *PhD Thesis Dissertation at University of Maryland College Park*, 2004.



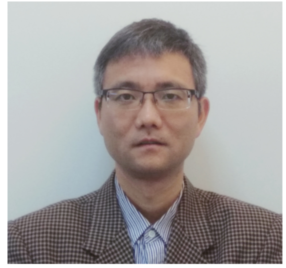
Menghai Pan received his B.S. degree in Computer Science from Huazhong University of Science and Technology, Wuhan, China, in 2016. He is a Ph.D. student in Computer Science at Worcester Polytechnic Institute.



Yanhua Li (S'09-M'13-SM'16) received two Ph.D. degrees in computer science from University of Minnesota at Twin Cities in 2013, and in electrical engineering from Beijing University of Posts and Telecommunications in China in 2009, respectively. He is currently an Assistant Professor in the Department of Computer Science at Worcester Polytechnic Institute (WPI) in Worcester, MA. Prior to this, he worked in Bell Labs in New Jersey, and Microsoft Research Asia in Beijing. His research interests are big data analytics and artificial intelligence in many contexts, including urban intelligence, smart cities, and urban planning and optimization. He is a recipient of NSF CAREER award and NSF CRII award.



Zhi-Li Zhang (M'97-SM'11-F'12) received the B.S. degree in computer science from Nanjing University, Jiangsu, China, in 1986, and the M.S. and Ph.D. degrees in computer science from the University of Massachusetts Amherst, Amherst, in 1992 and 1997, respectively. In 1997, he joined the Computer Science and Engineering faculty at the University of Minnesota, Minneapolis, MN, where he is currently a Professor. From 1987 to 1990, he conducted research with the Computer Science Department, Aarhus University, Aarhus, Denmark, under a fellowship from the Chinese National Committee for Education. He has held visiting positions with Sprint Advanced Technology Labs, Burlingame, CA; IBM T. J. Watson Research Center, Yorktown Heights, NY; Fujitsu Labs of America, Sunnyvale, CA; Microsoft Research China, Beijing, China; and INRIA, Sophia-Antipolis, France.



Jun Luo is a principal researcher at Lenovo Machine Intelligence Center, Lenovo Group Limited, in Hong Kong. He received his PhD degree in computer science from the University of Texas at Dallas, USA, in 2006. His research interests include big data, machine learning, spatial temporal data mining and computational geometry. He has published over 90 journal and conference papers in these areas.