Smart Cloud Commuting with Shared Autonomous Vehicles: A First Feasibility Study

Menghai Pan, Yanhua Li
Worcester Polytechnic Institute(WPI)
Worcester, Massachusetts
{mpan,yli15}@wpi.edu

Taihui Li, Zhi-Li Zhang
University of Minnesota, Twin Cities
Minnesota
{lixx5027,zhang089}@umn.edu

Jun Luo
Lenovo Machine Intelligence Center
Hong Kong, China
jluo1@lenovo.com

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 system 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.

KEYWORDS

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.

The emergence of autonomous vehicles[1] (AVs) 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 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 system 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

[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.

UrbComp'18, August 20, 2018, London, UK.

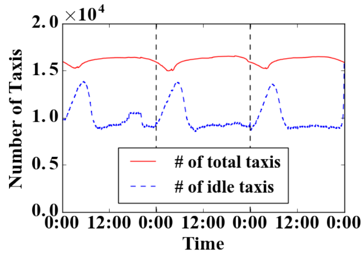


Figure 1: Idling taxis

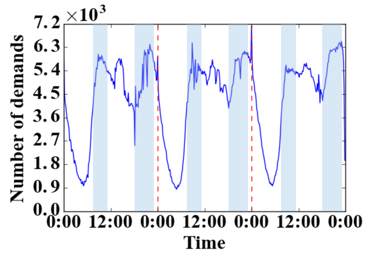


Figure 2: Request pattern

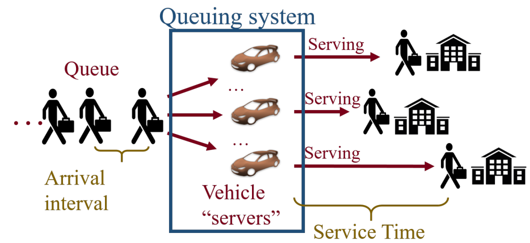


Figure 3: Queuing system

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 system. 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 that 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). Similar to today's (mobile-app-based) ride-hailing services, 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. 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. In other words, the AVs in SCCS cannot be replaced by the vehicles with drivers like taxis and Uber cars. One key difference between AVs and taxis or Uber cars is that each taxi or Uber car with a driver is maximizing its own gain, and each taxi/Uber car acts as a selfish player without caring much on the global gain in terms of transit system efficiency. etc. While with AVs, SCCS system can be designed to maximize a global system efficiency.

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 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). Notice that SCCS does not necessarily require the presence of depots. In this paper, we explore the trade-off between the centralized and decentralized SCCS system design. Clearly, current system design like Uber is using a fully decentralized approach. We use the number of depots as a parameter to control the trade-off and evaluate the system effiency gain. When the number of depots is sufficiently large, it becomes a decentralized system. 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.3 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.

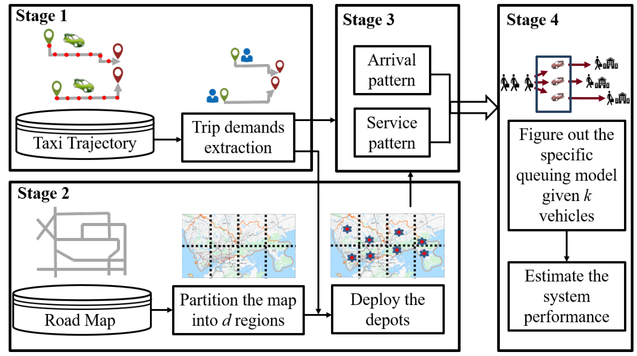


Figure 4: Framework

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. 4 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 demand 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 (Depot 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.

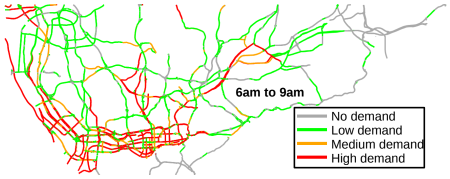


Figure 5: Heat map of starting location

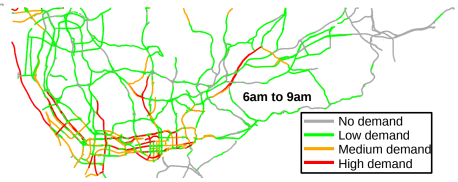


Figure 6: Heat map of ending location

Figure 7: Shenzhen road map

- **Stage 3 (Arrival/service pattern extraction)** With a particular SCCS system 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.

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 with GPS sets, where each GPS set generates a GPS point every 40 seconds on average. 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.

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 in Fig.7.

3.3 Stage 1: Demand 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 [8] 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.

After the two steps, we obtain a total of $595,501$ daily trip demands from our trajectory data. Fig.5 and Fig.6 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: Depot 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

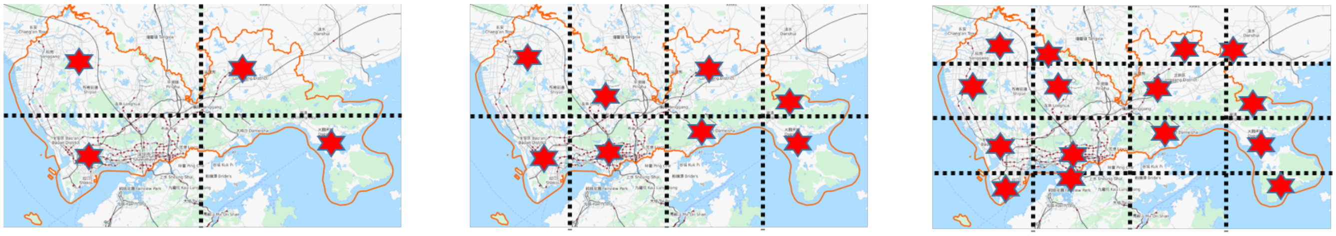


Figure 8: Depot placement in Shenzhen

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

locations is not on the road network, it will be shifted to the nearest road segment. Fig.8 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 $k_i = k \cdot N_i/N$.

3.5 Stage 3: Arrival/Service Pattern Extraction

SCCS ststem 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. The arrival rate distribution from original data can be nicely fitted by Poisson distribution. The parameter λ of Poisson distribution is the mean arrival rate, which is listed in Table 2 for different time intervals in a day.

Service pattern analysis. As shown in Fig.9, 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)[2].

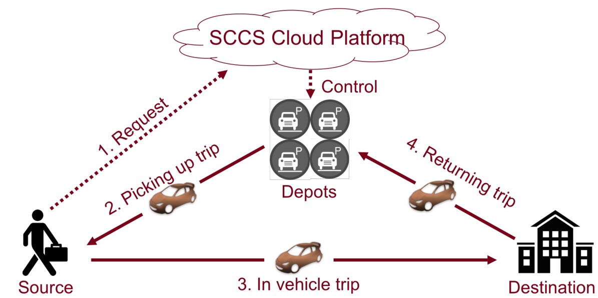


Figure 9: Service process in SCCS

Table 3: Average service time

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

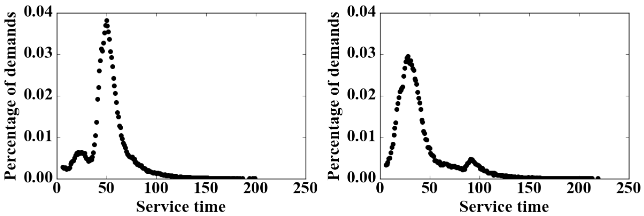
Since each request will be served by a vehicle from the depot in the source region, and the destination of the demand may be in a different region, a vehicle balancing approach is required. We adopt a simple schedule-based approach for vehicle rebalancing: Every 12 hours, the vehicles will be rebalanced 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.10 show the distributions of service time given different number of depots: 2 and 4 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 distribution of the service time from the trip demands data cannot be fitted by 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.

[2]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.



(a) 2 depot (b) 4 depots

Figure 10: Service time($k = 12000$)

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 [9]. We employ the approximation algorithm [11] 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).

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

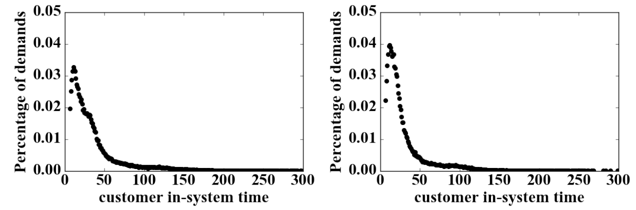
where ρ is the utilization in a queuing system, which equals to $\lambda\overline{S}$, and $Er_c(k, \rho)$ is the Erlang C formula(Eq (3)), which indicates the probability that an arriving customer has to wait, which is also the proportion of time that all k servers are busy. $k > \rho$ ensures the system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

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

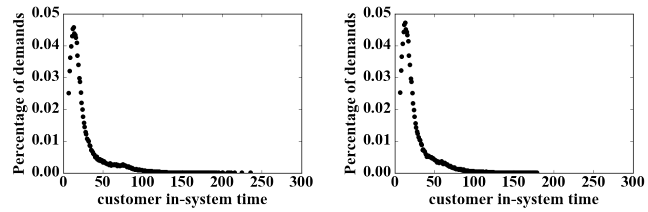
Finally, we can approximate the average waiting time in $M/G/k$ queue. Taking one depot deployment as an example, the arrival rate in $12pm - 6pm$ slot is 5.0594, and the average service time of the system is 3536.45249, so the utilization $\rho = 17876.4137$, and the coefficient of variation of the 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

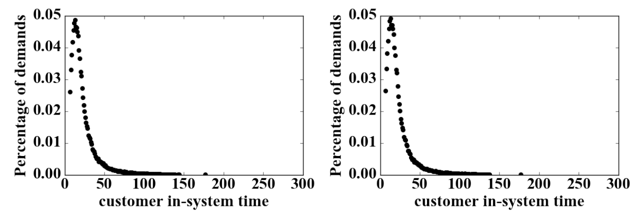
In this section, we use real taxi trip data to conduct experiments to evaluate (1) the performance of the design choices of number of available vehicles k and the number depots d. (2) the efficiency gain in SCCS comparing with current taxi system.



(a) 9000 vehicles (b) 10000 vehicles



(c) 11000 vehicles (d) 12000 vehicles



(e) 15000 vehicles (f) 20000 vehicles

Figure 11: the impact of total number of taxis

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 system (in different design choices) with the current taxi system. To evaluate how our SCCS system 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[3] (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 system, and get the passenger in-system time, which is shown in Fig 11.We can observe that as we increase the number of vehicles, the passenger in-system time decreases.

Moreover, Fig. 12 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.

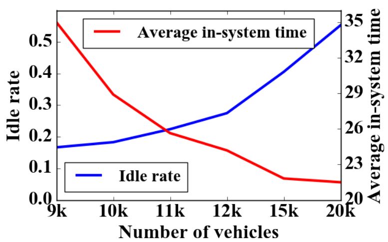


Figure 12: Tradeoff

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

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.

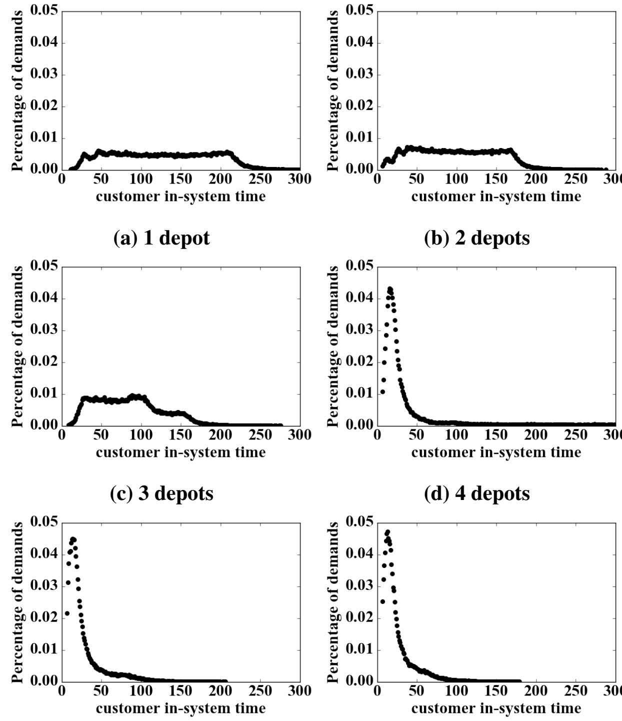
Fig. 12 clearly indicates the trade-off between the waiting time and the idle rate when changing the number of vehicles.

The number of depots in our system can also have effects on the customer's experience. Taking $k = 12000$ for example, Fig. 13 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. 13(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. 14–15 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.

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.

4.3.1 Vehicle utilization. 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.



(a) 1 depot (b) 2 depots (c) 3 depots (d) 4 depots (e) 8 depots (f) 16 depots

Figure 13: The impact of number of depots (k=12,000)

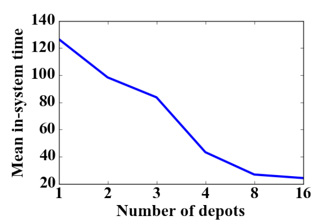


Figure 14: Average in-system time

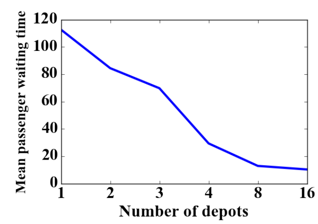


Figure 15: Average passenger waiting time

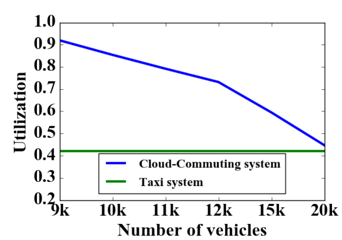


Figure 16: utilization of vehicles

[3]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.

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

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

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. 16 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 $T_{in-system}$ and the vehicle idle rate R_{idle} (see Fig. 12). Notice that the $T_{in-system}$ implies the Quality of Experience (QoE) of a passenger. The smaller $T_{in-system}$ is, the better QoE is. On the other hand, R_{idle} indicates the system operation cost. A large R_{idle} infers a low utilization of vehicles, which will increase system operation cost, thus the passengers' travel cost. So 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 $[0, 1]$, which captures the design trade-off between the passenger QoE and passenger travel cost. The smaller V-value indicates better performance. Taking 32 depots and $\alpha = 0.01$ as an 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.

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 ([4, 15, 19–21, 23, 27, 29]) 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 [20], 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 [27] and [21], 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. 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 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 [5–7, 10, 12–14, 16–18, 22, 24–26, 28]. For examples, [12] 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. [22] develops novel models to predict future crowd flow traffic in subway stations. [24] introduces a method to estimate the travel time in a road segment using sparse trajectories data. [26] proposes a model to discover urban function zones by exploring latent activity trajectory data. In [28], 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 ACKNOWLEDGEMENTS

Yanhua Li was supported in part by NSF CRII grant CNS-1657350 and a research grant from Pitney Bowes Inc. Taihui Li and Zhi-Li Zhang were supported in part by US DoD DTRA grant HDTRA1-14-1-0040 and NSF grants CNS-1411636, CNS-1618339 and CNS-1617729.

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. Bao, T. He, S. Ruan, Y. Li, and Y. Zheng. Planning bike lanes based on sharing-bikes' trajectories. August 2017.

[6] 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.

[7] 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.

[8] H.-P. Hsieh, S.-D. Lin, and Y. Zheng. Inferring air quality for station location recommendation based on urban big data. In *SIGKDD*, 2015.

[9] J. Kingman. The first erlang century—and the next. *Queueing systems*, 63(1-4):3, 2009.

[10] C. Kumar, D. Basu, and B. Maitra. Modeling generalized cost of travel for rural bus users: a case study. *Journal of Public Transportation*, 2004.

[11] A. Lee and P. Longton. Queueing processes associated with airline passenger check-in. *OR*, pages 56–71, 1959.

[12] 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.

[13] 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.

[14] C. Liu, K. Deng, C. Li, J. Li, Y. Li, and J. Luo. The optimal distribution of electric-vehicle chargers across a city. In *IEEE International Conference on Data Mining*, pages 261–270, 2017.

[15] G. Liu, M. Pan, Y. Li, Z. L. Zhang, and J. Luo. Modeling urban trip demands in cloud-commuting system: A holistic approach. In *Computer Communications Workshops*, pages 857–862, 2017.

[16] X. Liu, X. Kong, and Y. Li. Collective traffic prediction with partially observed traffic history using location-base social media. In *CIKM*, 2016.

[17] X. Liu, X. Kong, and Y. Li. Collective traffic prediction with partially observed traffic history using location-based social media. In *ACM International on Conference on Information and Knowledge Management*, pages 2179–2184, 2016.

[18] 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.

[19] M. Maciejewski, J. Bischoff, and K. Nagel. An assignment-based approach to efficient real-time city-scale taxi dispatching. *IEEE Intelligent Systems*, 2016.

[20] 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*, 2015.

[21] M. Pavone. Autonomous mobility-on-demand systems for future urban mobility. In *Autonomous Driving*, pages 387–404. Springer, 2016.

[22] 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.

[23] 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.

[24] 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.

[25] G. Wu, Y. Ding, Y. Li, J. Bao, Y. Zheng, and J. Luo. Mining spatio-temporal reachable regions over massive trajectory data. In *IEEE International Conference on Data Engineering*, pages 1283–1294, 2017.

[26] 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.

[27] 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.

[28] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, and E. Chang. Diagnosing New York City's Noises with Ubiquitous Data. In *Ubicomp*, 2014.

[29] 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.