Mobile Netw Appl
DOI 10.1007/s11036-016-0675-y

Transfer Problem in a Cloud-based Public Vehicle System with Sustainable Discomfort

Ming Zhu[1] · **Xiao-Yang Liu**[1] · **Meikang Qiu**[2] · **Ruimin Shen**[1] · **Wei Shu**[3] · **Min-You Wu**[1]

Abstract The increasing population in urban areas gives rise to a huge traffic pressure. A cloud-based industrial system, public vehicle (PV) system, is promising to mitigate the traffic congestion in smart cities, where passengers can share PVs and transfer among them with scheduling decisions made by the cloud. This paper studies the transfer problem in the PV system due to that transfer can improve the whole traffic efficiency with sacrificing a little comfort with the corporation of all the PVs. The transfer problem is NP-Complete through our analysis. Our work can be separated into three steps. First, we introduce several factors to guarantee the comfort of passengers during transfer. Second, we propose two algorithms through the graph-based scheduling problem aiming at reducing the travel distance of all the PVs with service guarantee. Third, simulations based on the Shanghai (China) urban road network show that, the total travel distance of PVs is reduced under the quality of service for passengers, and the traffic efficiency is improved.

Keywords Public vehicle · Transfer · Multi-hop ridesharing · Vehicular networks

1 Introduction

A public vehicle (PV) system [1] is a cloud-assisted industrial system [2] in smart cities with several constraints aiming at improving transportation resource sharing for passengers and increasing the whole traffic efficiency. The cloud devises scheduling strategies and paths for PVs through demands of passengers, and the travel demands of passengers can be satisfied.

The PV system consists of three parts: a cloud center, PVs, and passengers. If one passenger needs trip service, she sends a request (including her origin, destination, and earliest start time, etc.) through a smart phone to the cloud. This cloud schedules PVs to serve her and calculates the paths of the PVs. Finally the corresponding PVs travel to pick her up at her origin and then drop her off at her destination. She can access the information of corresponding PVs through apps, e.g., vehicle ID, locations, speed, and paths.

The PV system has several advantages, e.g., lower price, high quality of service, compared with the current transportations. On one hand, the PV system provides lower price than private cars and taxis with a little discomfort due to more traffic sharing. Besides, there is no parking cost. On

---

✉ Ming Zhu
zhumingpassional@sjtu.edu.cn

✉ Xiao-Yang Liu
yanglet@sjtu.edu.cn

Meikang Qiu
mqiu@pace.edu

Ruimin Shen
rmshen@sjtu.edu.cn

Wei Shu
shu@ece.unm.edu

Min-You Wu
mwu@sjtu.edu.cn

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China

[2] Department of Computer Science, Pace University, New York, NY, USA

[3] Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA

the other hand, the PV system provides higher service quality than buses. Buses have to run even without passengers with fixed routes, while the PVs run under the scheduling strategies of the cloud.

For transportation systems, the key to achieve high traffic efficiency is ridesharing, which allows passengers to share vehicles in common segments of their itineraries. The ridesharing in the PV system is different from traditional ridesharing systems, e.g., Uber Pool (https://www.uber.com/) or Lyft (https://www.lyft.me/). Traditional ridesharing (also referred as carpool, or taxi-sharing) is a distributed system, and the paths are determined by drivers and riders under the competition or tradeoff among them. Once one ridesharing has been conducted, some newly generated requests will be neglected even they perfectly match it. However, in the PV system, the ridesharing strategies and paths are calculated by the cloud, and the newly generated requests can be served if only there are available seats, making the ridesharing more dynamic and more flexible. The relationship between PVs is to corporate to achieve better travel service, and there is no competition between them.

To further improve the traffic efficiency of the PV system, we study the transfer problem, which is also named as multi-hop ridesharing in some other papers [3]. Transfer may provide better service, and lower trip price for passengers due to more flexibility. Teubner et al. analyze the structure and the economics of electronic ride sharing markets [4], and find that multi-hop ridesharing proves competitive against other transportation systems and has the potential of greatly improving ride availability and city connection. Transfer in the PV system makes the scheduling mechanism more complex.

We study the transfer problem in the PV system to reduce the total travel distance with service guarantee for passengers. In current solutions, the transfer points are predicted by the paths of vehicles, or are got by the tradeoff between drivers and passengers. Moreover, the comfort of passengers should not be neglected. If the service quality of passengers can be guaranteed, the lower travel price makes more passengers accept transfers.

To the best of our knowledge, this work is the first one to consider the transfer problem in the PV system. We study this problem in a practical setting by exploiting the downtown road network of Shanghai. The contribution of this paper is separated into three parts.

- We propose the transfer problem of the cloud-based PV system, and analyze its NP-Completeness. Then we introduce several factors to guarantee the comfort of passengers.
- We propose two algorithms aiming at reducing the travel distance of all the PVs with service guarantee for passengers, e.g., detour ratio, transfer time, maximum transfer times. Therefore, the gasoline consumption and carbon emission are saved.
- We build simulations in Shanghai (China) urban road network to evaluate the performance of proposed algorithms. The total travel distance of the PVs is reduced under service guarantee, and the number of moving vehicles on roads decreases, therefore the traffic congestion is mitigated.

The related work about transfers is described in Section 2. Section 3 details the scenario of the PV system. In Section 4, we describe the transfer problem in the PV system. Section 5 details proposed algorithms. Section 6 shows the performance of proposed algorithms. Section 7 presents the conclusions and future work about this problem.

2 Related work

Transfer problem in this paper is one type of ridesharing where passengers can transfer between vehicles. Ridesharing has drawn a lot of attention from researchers. Ridesharing can be roughly divided into two categories: one-hop ridesharing [5] and multi-hop ridesharing [3]. Semantically, in one-hop ridesharing systems, each passenger is served by only one vehicle without transfer, who will be picked up in her origin and dropped off in her destination by the same vehicle. However, in multi-hop ridesharing systems, one passenger can transfer between different vehicles and may be served by multiple vehicles. Obviously, multi-hop ridesharing is more flexible than one-hop ridesharing, and can make better use of vehicle capacities, but it reduces the comfort of passengers. Transfers in ridesharing systems may be accepted by many people for the following reasons. Some research shows that, about 45 % of people can accept ridesharing with others [6], and we believe the lower price can attract some people to accept transfers. The information communication technology (ICT) makes personal trip service more convenient. With cloud-assisted computing [7] or mobile cloud computing [8, 9], it is common for people to access or share their information on the cloud using mobile devices, although some privacy problems [10] need to be studied.

Most ridesharing is one-hop ridesharing [11]. The mobility profiles of individuals from raw digital traces (e.g., GPS traces) reflect the behavior the drivers or riders [12], which can be used to develop a carpool matching that satisfies various basic constraints. Zhang et al. have proposed one solution for carpool [13]. Passengers should gather at one origin (e.g., airport) to start a carpool, which limits the expansion of this system.

Less research is carried out on multi-hop ridesharing [3], which is also named as ridesharing with transfers [14].

Ray et al. have quantified the critical mass or tipping-point in the number of drivers offering seats for a casual or dynamic ridesharing line to work, and they find that ridesharing with transfers can be used to reach this critical mass [15]. Multi-hop ride sharing has the potential to largely improve the connection of cities and availability of ride, especially under high reliability requirements. The multi-hop ridematching problem [16] provides more flexibility than other forms of ridematching strategies and it proposes more choices for users. The solutions of dynamic pickup and delivery with transfers [17] can only provide possible transfer points with costs constraints, which are calculated through tradeoff of vehicles, and some solutions do not consider the detour distance of passengers. The dial a ride problem with transfers [18] aims at designing paths for user requests with several constraints, e.g., precedence between pickup points (origins) and delivery points (destinations), time windows.

The above solutions concerning transfers should consider the acceptance, detour distance, transfer time, price for passengers, and passengers and drivers should compromise for different interests. For example, drivers want to earn more money by serving more passengers, and passengers would be dissatisfied about too much detour. In traditional ridesharing systems, there exist a lot of competitions or tradeoffs between drivers, and game theory between drivers and riders. However, the relationship between PVs is to corporate with each other to provide better service, and the feedback for drivers of PVs is calculated by the cloud, which considers of the fairness, cost, labor, etc. Moreover, in the PV system, each PV can serve new passengers once it has available seats, and this is different from traditional ridesharing systems, which makes the transfer problem different from the scenarios of the existing solutions.

3 PV system

In this section, we first describe the architecture of the PV system, and then compare travel costs of the PV system with other transportation systems.

3.1 Architecture

Figure 1 shows the architecture of the PV system, which includes the following three parts, a cloud center, PVs, and passengers. PVs are one type of electric vehicles, and the scheduling for charging [19] is an important issue. The solid lines denote the communication between them, and the dash lines imply the scheduling tasks of PVs. The right-hand PV is scheduled to serve two passengers. Vehicle to vehicle communication (V2V) is supported in vehicular ad-hoc networks (VANETs), therefore PVs can communicate with each other within transmission range. V2V can enhance the

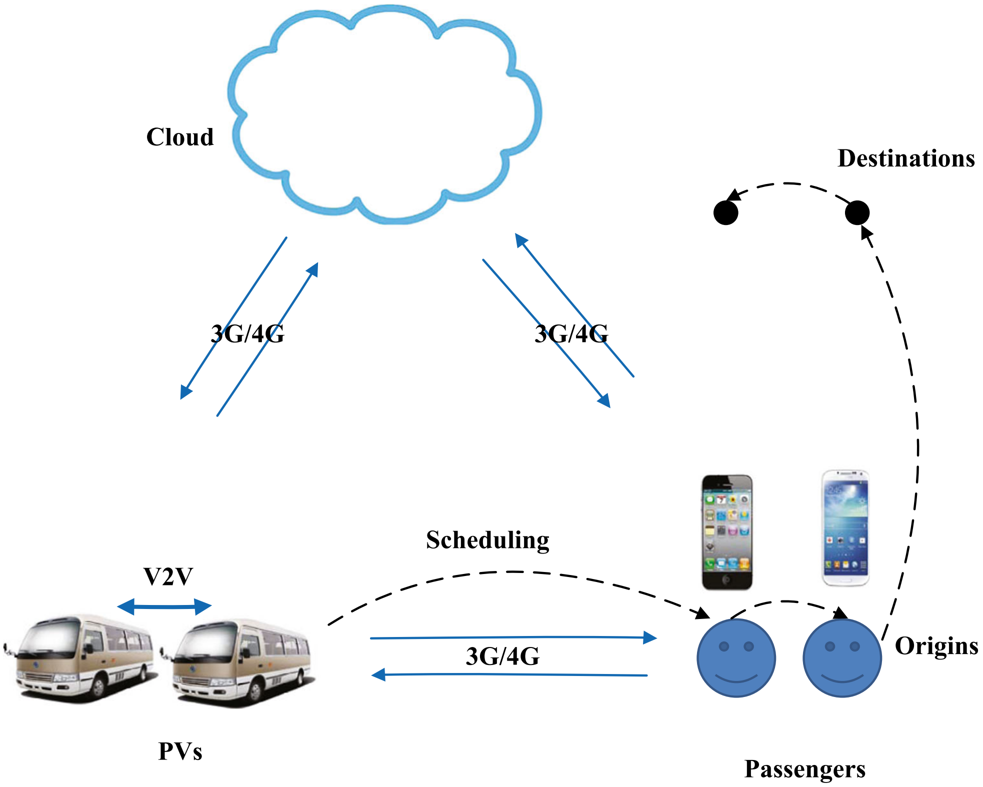


Fig. 1 Architecture of the PV system

corporation of PVs, e.g., emergency alert. The design for paths and scheduling tasks for PVs is named as PV path (PVP) problem, which is NP-Complete, and one solution for it has been designed [1]. Obviously, the transfer problem is NP-Complete by adding transfers for passenger based on PVP. The scheduling decisions in transfer problem are calculated by the cloud, and then they will be executed by PVs.

3.2 Cost comparison

The price by public transportation systems in some large cities is shockingly low for the subsidization of governments. The costs by taxis are round six times of that by buses in Hong Kong [20]. Table 1 shows the transportation fares of several large cities in the world. The price of each km ride of public transportation (bus, subway, tram, or metro) and taxi is presented in the second and third columns. We know that, in Shanghai, the price by taxi per km is about 10 times of that by subway/metro.

The costs of current ridesharing systems are still high. Some survey results show that, UberX is about 40 % cheaper than traditional taxis on average [21]. We believe that, the costs by the PV system will be less than Uber Pool for its flexibility and dynamic. Moreover, if passengers can transfer between PVs, the costs will be further reduced.

4 Transfer problem

In this section, we mainly introduce the transfer problem in the cloud-based PV system.

Table 1 Transportation fares in the world

City	Public transportation (per km)	Taxi (per km)
Shanghai	\$0.20 (metro)	\$2
Tokyo	\$0.33 (metro)	\$2.8
Bangkok	\$0.19 (skytrain, subway)	\$2
Sydney	\$0.47 (metro, bus)	\$1.5
Auckland	\$0.20 (bus, train)	\$3
Barcelona	\$0.28 (tram, bus, metro)	\$2.1
Rome	\$0.20 (tram, bus, metro)	\$2.7
Athens	\$0.20 (tram, bus, metro)	\$2.6
Berlin	\$0.42 (tram, bus, metro)	\$3.5
New York	\$0.33 (subway, bus)	\$2
Los Angeles	\$0.22 (bus, metro)	\$1.8
San Francisco	\$0.29 (tram, bus, metro)	\$2.2
New Orleans	\$0.22 (tram, bus)	\$1.3

If passengers can transfer with sustainable detour and delay in the PV system, we have to consider the effects of transfer, including the change of paths of PVs, the comfort of other passengers, and the whole traffic efficiency. Hou et al. have pointed out that, more than one transfer does not bring any noticeable benefits [22]. In addition, considering of the comfort, in this paper we assume that each passenger can transfer at most once. The solutions we proposed can also be used to multiple transfers for each request by adding more iterations.

Let $p \in P$ denote one PV, and $r \in R$ denote one request or passenger (one request corresponds to one passenger). A request r can be denoted by (r_o, r_d, r_e), where r_o is her origin, and r_d is her destination, and r_e is the earliest start time. If some PVs are scheduled to r, she will be picked at r_o, and dropped at r_d. Let r_t denote the transfer point of r. If r will transfer from p_1 to p_2 at r_t, the paths of the two PVs may change, and the service quality of requests two PVs serve may also change. In this paper, we temporarily do not

Table 2 Denotations

R	set of requests
m	number of requests. $m = \|R\|$
P	set of PVs
n	number of PVs. $n = \|P\|$
$q(p)$	path of PV p
$q(r)$	path of request r
$d(p)$	travel distance of one PV p. $d(p) = \|q(p)\|$. Unit: km
$d(r)$	travel distance of one request r. $d(r) = \|q(r)\|$. Unit: km
$d_s(r)$	shortest distance from origin r_o to destination r_d of request r. Unit: km
$d(i, j)$	shortest distance from i to j
$b(p)$	last location of $q(p)$
$N_{max}(p)$	maximum number of passengers in p at any location on its path
c	capacity of PVs
$t(p, i)$	arrival time of p at location i
$\delta(r)$	transfer period. Unit: minute
Δ	threshold of transfer period. Unit: minute
$\theta(r)$	detour ratio of r
$\bar{\theta}$	average detour ratio
Θ	threshold of detour ratio
$l(p)$	service list, which denotes a set of requests p is serving or will serve
d_{min}	minimum distance from origin to destination for any request. Unit: km
e	speed of PVs. Unit: km/h
α	the ratio of passengers to PVs

consider the latest arrival time. If the latest arrival time is considered, it is hard to compare different solutions due to that, some requests may be rejected by the PV system, and in Section 6, we will evaluate the performance of proposed solutions.

Some variables are shown in Table 2. $\delta(r)$ denotes the transfer period, i.e., the period between the dropoff time from one PV and the pickup time by another PV. $\theta(r)$ is detour ratio, which denotes the percentage of additional travel distance compared with $d_s(r)$. $\theta(r) = (d(r) - d_s(r))/d_s(r)$. Make sure that, $\delta(r)$ does not exceed a threshold Δ and $\theta(r)$ does not exceed a threshold Θ. The average detour ratio of all requests is denoted by $\bar{\theta} = (\sum_r d(r) - \sum_r d_s(r))/\sum_r d_s(r)$. With respect to any request, ensure that the shortest distance from its origin to destination is not shorter than d_{min}.

We assume that all the paths of PVs have been calculated by the no transfer solution in [1], and then we study the transfer strategies based on the previous work. To attract the passengers to accept transfer, we can lower the trip price and provide service within sustainable discomfort, e.g., low detour, and short transfer period. It is a pricing problem and is not in the scope of this paper. We assume that all the passengers in the PV system accept transfers under service guarantee, and any PV should not wait for passengers. If there are several passengers in this PV, some of them may be upset about waiting for even one minute.

Make sure that, any transfer is conducted under some service constraints with preserving the comfort of passengers, otherwise, transfer is not allowed. With respect to r' who will transfer from p_1 to p_2, p_1 is named as a sender (denoted by S), and p_2 is named as a receiver (denoted by V). $q(p_1)$ and $q(p_2)$ are the initial paths of p_1 and p_2 calculated by the no transfer solution. Let $q'(p_1)$ and $q'(p_2)$ be the new paths of p_1 and p_2 calculated by transfer solutions. $q'(p_1)$ is got based on $q(p_1)$: Remove r'_d and insert r'_t with r'_o preceding r'_t. $q'(p_2)$ is got based on $q(p_2)$: Insert r'_t and r'_d with r'_t preceding r'_d. The objective function of the transfer problem is to minimize the total travel distance of PVs:

$$\text{Objective: } \min \sum_{p \in P} d(p) \tag{1}$$

Let $L(r') = \{r'\} \bigcup l(p_1) \bigcup l(p_2)$. For any $r \in L(r')$, and any PV, the service of quality (QoS) constraints should be obeyed:

- (a) $N_{max}(p) \leq c,\ p \in P$
- (b) $r'_t \in \{q(p_1) \bigcap q(p_2)\}$.
- (c) $t(S, r_o) < t(S, r_t) < t(V, r_t) \leq t(V, r_d)$.
- (d) $\delta(r) = t(V, r_t) - t(S, r_t) \leq \Delta$.
- (e) $\theta(r) \leq \Theta$.

(**a**) indicates the capacity constraint. The maximum number of passengers at any location should not exceed the capacity of PVs. (**b**) implies that r'_t is one intersection on paths of p_1 and p_2, which makes sure that both p_1 and p_2 traverse through the transfer point r'_t. (**c**) implies the arrival time constraints of the two PVs at r_o, r_t, and r_d. The sender S should arrive at the transfer point before the receiver V. (**d**) means the transfer period should not exceed its threshold, which is between the time r is dropped off by S, and the time r is picked up by V. (**e**) implies the detour ratio should not exceed its threshold. The above constraints make sure that, passengers can accept transfer with sustainable discomfort, e.g., transfer period, detour. The service quality of r if she transfers is denoted $f(r)$, where w_1 and w_2 indicate weights.

$$f(r) = w_1 * \theta(r) + w_2 * \delta(r) \leq w_1 * \Theta + w_2 * \Delta \tag{2}$$

If the QoS of all requests is guaranteed, the travel distance reduction becomes an important issue, and we choose the transfer strategy which produces the maximum distance reduction: $\pi(r', r'_t) = q(p_1) + q(p_2) - q'(p_1) - q'(p_2) > 0$.

Figure 2 indicates one transfer case. There are two PVs, p_1 and p_2. The origins and destinations of three passengers A, B, and C are denoted by circles and squares respectively. The initial path (without transfer) of p_1 is denoted by the solid black line, and the path of p_2 or passenger C is denoted by the red solid line. $q(p_1) = q(B) = \{B_o \to k \to A_d \to B_d\}$. $q(p_2) = q(C) = \{C_o \to k \to C_d\}$. If B transfers from p_1 to p_2 at the intersection k, the new path (with transfer) of p_1 is $q'(p_1) = \{B_o \to k \to A_d\}$, which is denoted by

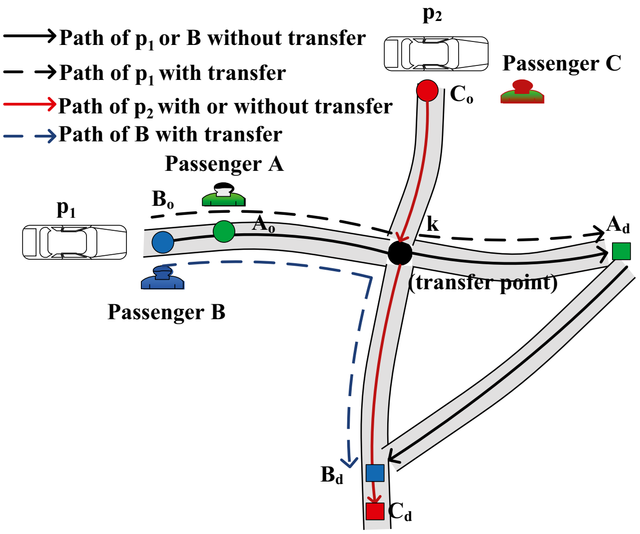
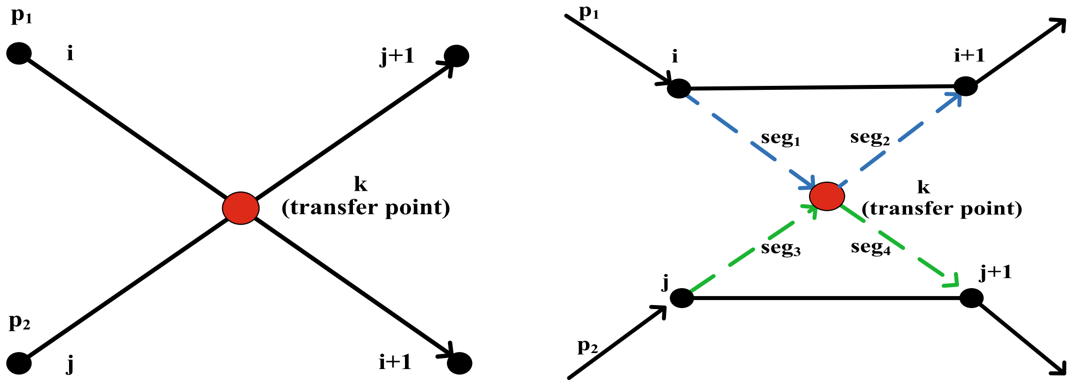


Fig. 2 Transfer case



(A) initial paths have at least an intersection (B) initial paths have no intersections

Fig. 3 Transfer between two segments of paths

the dash black line, and the paths of p_2 and passenger C do not change ($q'(p_2) = q(p_2)$), and the new path of B is $q'(B) = \{B_o \to k \to B_d\}$, which is denoted by the dash blue lines. We can see that, the reduced travel distance of p_1 is $d(p_1) - d'(p_1) = d(A_d, B_d)$, and the travel distance of B is also reduced, because $d(B) - d'(B) = d(k, A_d) + d(A_d, B_d) - d(k, B_d) > 0$. We know that, the total travel distance of PVs, and the travel distance of passengers both can be reduced through transfer.

5 Solutions

In this section, we first introduce method of searching transfer points for one request between two PVs, and then we propose two algorithms to search transfer points for all requests. All the solutions are computed by the cloud and then the corresponding PVs travel under the commands of the cloud.

5.1 Searching transfer points for one request

On the path of any PV p, the origins, destinations, or transfer points of requests this PV serving or will serve are named as *anchors*, which divide the path to multiple segments. Here, we first consider calculating the transfer point among two segments.

We assume that one request r can transfer from p_1 to p_2. If one point k is selected as a transfer point, and the QoS of all requests is guaranteed, k is named as a *compatible* transfer point. Let a function *COMPATIBLE(t)* check that if one point t is compatible, which returns $true$ or $false$. Let $\rightharpoonup i, j$ denote the shortest path from i to j. In Fig. 3, i and $i+1$ are two consecutive anchors on path of p_1, and j and $j+1$ are two consecutive anchors on path of p_2.

– Case A: initial paths of p_1 and p_2 have at least one intersection.
  Figure 3a shows this case and the intersection is denoted by a red point k. If k is compatible, it will be chosen as a transfer point. Otherwise, we have to try four anchors (i, $i+1$, j, and $j+1$) and check if they are compatible. Finally, record the compatible transfer points.
– Case B: initial paths of p_1 and p_2 have no intersections.
  Figure 3b shows this case. If $\rightharpoonup i, j+1$ and $\rightharpoonup j, i+1$ have at least one intersection, and r can transfer at one compatible point k, clearly, k is the best transfer point, because in any other points, the travel distance of two PVs is not shorter than it. If none of intersections is compatible or $\rightharpoonup i, j+1$ and $\rightharpoonup j, i+1$ do not have any intersection, we try four anchors (i, $i+1$, j, and $j+1$) and check if they are compatible. Finally, record the compatible transfer points.

Algorithm 1 shows the searching compatible transfer points (SCTP) algorithm for one request r if she transfers from p_1 to p_2. Let T denote compatible transfer points of r. Lines (1–6) are initialization, through which each anchor on path of p_1 and p_2 will be checked to see if it is compatible. Lines (8–25) calculate the transfer point between each segment of paths of two PVs with two cases as we have described. In line (10), K_1 is a set to store transfer points we have calculated.

After compatible transfer points are obtained, we should insert r_d (destination of r) on path of p_2 and get the new path. Obviously, make sure that r_t precedes r_d. Let $\{Q, D\} = INSERT(r_d, q(p_2))$ denote the paths and corresponding distance if we insert r_d on path of p_2 with r_t preceding r_d, where $Q = \{q'(p_2)\}$ is a set of paths and $D = \{d'(p_2)\}$ is a set of distances of corresponding paths.

With respect to r, let Π, Q^1, and Q^2 be three sets which denote the reduced travel distance, new paths of p_1 and p_2 respectively. Π_i, Q_i^1, and Q_i^2 denote the i^{th} element in corresponding sets. Let Γ be a set of multiple receivers, and let Λ be a set of transfer points. Here, the reduced travel distance is the distance of paths without transfer minus the one with transfer.

```
Algorithm 1 Searching compatible transfer points (SCTP)
Input:
    $r, p_1, p_2$
    $\{q(p_1), q(p_2)\}$: initial paths of $p_1$ and $p_2$ calculated
    by the no transfer solution [1]
Output:
    $T$: compatible transfer points of $r$
$T \leftarrow \emptyset$;
for each anchor $k \in \{q(p_1) \bigcup q(p_2)\}$ do
    if COMPATIBLE(k) = true then
        $T \leftarrow T \bigcup \{k\}$;
    end if
end for

for anchor $i \in \{q(p_1) \setminus b(p_1)\}$ after $r$ is picked do
    for anchor $j \in \{q(p_2) \setminus b(p_2)\}$ do
        $K_1 \leftarrow \emptyset$;
        if $K_2 \leftarrow \{\overrightarrow{i, i+1} \bigcap \overrightarrow{j, j+1}\} \neq \emptyset$ then
            $K_1 \leftarrow K_2$;
        else
            Calculate $\overrightarrow{i, j+1}$ and $\overrightarrow{j, i+1}$;
            if $K_3 \leftarrow \{\overrightarrow{i, j+1} \bigcap \overrightarrow{j, i+1}\} \neq \emptyset$ then
                $K_1 \leftarrow K_3$;
            end if
        end if
        for $k_1 \in K_1$ do
            if COMPATIBLE($k_1$) = true then
                $T \leftarrow T \bigcup \{k_1\}$;
            end if
        end for
    end for
end for
return T
```

Algorithm 2 shows the searching best transfer point (SBTP) algorithm for one request r if she transfers from p_1 to p_2. The main idea of SBTP is that, for each compatible transfer point, insert r_d on path of p_2 with r_t preceding r_d and get the new path of p_2, and then choose the corresponding transfer point and paths which most reduce the travel distance of PVs with QoS guarantee.

```
Algorithm 2 Searching best transfer point (SBTP)
Input:
    $r, p_1, p_2$
    $T$: compatible transfer points of $r$ calculated by Algorithm 1
    $\{q(p_1), q(p_2)\}$: initial paths of $p_1$ and $p_2$ calculated
    by the no transfer solution [1]
Output:
    $r_t$: best compatible transfer point of $r$
    $\pi(r, r_t)$: maximum travel distance reduction of PVs
    $q'(p_1), q'(p_2)$: new paths of $p_1$ and $p_2$
 1: $r_t \leftarrow null$;
 2: $\{\Pi, Q^1, Q^2, \Lambda\} \leftarrow \emptyset$;
 3: for $k \in T$ do
 4:     $q''(p_1) \leftarrow q(p_1) \backslash \{r_d\}$;
        $d''(p_1) \leftarrow |q''(p_1)|$;
 5:     $\{Q, D\}$ = INSERT($r_d, q(p_2)$);
 6:     if $\pi = d(p_1) + d(p_2) - d''(p_1) - min\ D > 0$ AND
        $min\ D = D_i$ then
 7:         Put $\pi$, $q''(p_1)$, $Q_i$, and $k$ to $\Pi$, $Q^1$, $Q^2$, and $\Lambda$
            respectively;
 8:     end if
 9: end for
10: if $max\ \Pi = \Pi_i > 0$ then
11:     $r_t \leftarrow \Lambda_i$.
12:     $\pi(r, r_t) \leftarrow \Pi_i$;
13:     $q'(p_1) \leftarrow Q_i^1$;
14:     $q'(p_2) \leftarrow Q_i^2$;
15: end if
16: return $\{r_t, \pi(r, r_t), q'(p_1), q'(p_2)\}$
```

In Algorithm 2, lines (3–9) calculate the transfer point and paths of PVs which produce the maximum reduced distance. Line (4) calculates the new path and travel distance of p_1. Line (6) implies that, we calculate the maximum reduced travel distance with QoS guarantee: $max\{d(p_1) + d(p_2) - d''(p_1) - d''(p_2)\} = d(p_1) + d(p_2) - d''(p_1) - minD, d''(p_2) \in D$. Line (7) means that, the reduced travel distance, paths of two PVs, and the compatible transfer point are put to corresponding sets. Lines (11–14) indicate that, if we find a transfer point which most reduces travel distance of PVs, the corresponding transfer point, reduced travel distance, and paths of PVs can be calculated.

5.2 Single transfer algorithm

SCTP calculates multiple compatible transfer points for one request, and SBTP calculates the best transfer point for one request. Here, we introduce one algorithm which determines the transfer points of all requests, and the new paths of all PVs.

Algorithm 3 shows the detail of single transfer (ST) algorithm, which is based on SCTP and SBTP. Here, the input variable R is a set of requests in each PV which are sorted from the farthest destination to the nearest. If the request with farthest destination can transfer, the travel distance of PVs may be largely reduced compared with the others. ST calculates a transfer point for each request which can most reduce the travel distance of PVs. Let $\{q(p)\}$ and $\{q'(p)\}$ denote the initial paths and new paths of PVs. Let $\{r_t\}$ denote the transfer points of requests R. Let $\{l'(p)\}$ denote the new service lists of PVs. If r has transferred, $transferred$ is true, otherwise, is false. If r needs transfer, $needtransfer$ is true, otherwise, is false.

Algorithm 3 Single transfer (ST)

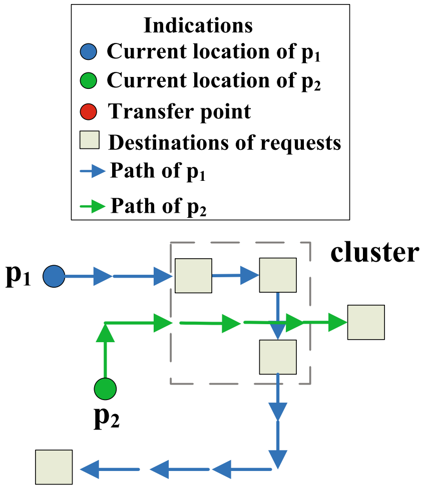
```
Input:
    $R$: set of requests
    $P$: set of PVs
    $\{q(p)\}$ $(p \in P)$: initial paths of PVs calculated by the no transfer solution [1]
Output:
    $\{r_t\}$ $(r \in R)$: transfer points of requests
    $\{q'(p)\}$ $(p \in P)$: new paths of PVs
    $\{l'(p)\}$ $(p \in P)$: new service lists of PVs
 1: for $r \in R$ do
 2:   if $transferred = false$ then
 3:     $\{\Pi, Q^1, Q^2, \Lambda, \Gamma\} \leftarrow \emptyset$;
 4:     for $V' \in \{P \backslash \{S\}\}$ do
 5:       Calculate $\{t', \pi(r,t'), q''(S), q''(V')\}$ by Algorithm 2;
 6:       if $t' \neq null$ then
 7:         Put $\pi(r,t')$, $q''(S)$, $q''(V')$, $t'$, and $V'$ to $\Pi$, $Q^1$, $Q^2$, $\Lambda$, and $\Gamma$ respectively;
 8:       end if
 9:     end for
10:     if $max\ \Pi = \Pi_i > 0$ then
11:       $V \leftarrow \Gamma_i$;
12:       $q'(S) \leftarrow Q_i^1$;
13:       $q'(V) \leftarrow Q_i^2$;
14:       $r_t \leftarrow \Lambda_i$;
15:       $l'(V) \leftarrow l'(V) \bigcup \{r\}$;
16:       $needtransfer \leftarrow true$;
17:     else
18:       $needtransfer \leftarrow false$;
19:     end if
20:   end if
21: end for
22: return $\{r_t\}$, $\{q'(p)\}$, $\{l'(p)\}$ $(r \in R,\ p \in P)$
```

In Algorithm 3, line (2) means that, if r has transferred, she does not need to transfer. Line (7) indicates that, if we can obtain a transfer point through Algorithm 2, we should put reduced travel distance of PVs, paths of sender and receiver, transfer point, and receiver to corresponding sets. Lines (11–15) imply that, we have found a transfer point which can most reduce traveling distance of PVs with QoS guarantee, and then the corresponding paths of PVs, transfer point, and service list of the receiver should be updated. Line (15) means that, V will serve r as a receiver. Line (18) denotes that, if we cannot find suitable transfer point with QoS guarantee, r does not need to transfer.

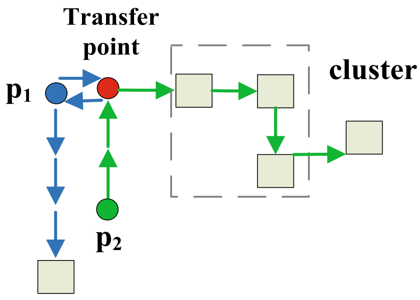
5.3 Cluster transfer algorithm

To improve the efficiency of transfer, we introduce another solution, cluster transfer (CT). The details are as follows. Cluster the requests with near destinations using single-linkage clustering. This cluster of requests will be dropped off together by one PV p_1, and then be picked up together by another PV p_2, and finally be dropped off by p_2 one after another at their destinations. We see that, in CT, the process of clustering is added based on ST. The steps except the cluster process are almost the same as ST. Therefore, we do not detail the steps of CT.

Figure 4 depicts the paths of two PVs with or without cluster transfer. Blue and green arrows denote the paths of p_1 and p_2 respectively. Blue and green circles denote the



(A) without cluster transfer.



(B) with cluster transfer.

Fig. 4 Paths of PVs (with or without cluster transfer)

Table 3 Values of variables

Variables	m	n	e	d_{min}	α	c	Θ	Δ
Values	50–250	50	40	5	1–5	16	0.3	4

current locations of p_1 and p_2, and squares denote destinations of requests who have been picked up, and the red point or cicle denotes a transfer point. Three requests/passengers (their destinations are surrounded by a dash square) who have been picked up by p_1 will transfer to p_2 at the red point. Each arrow denotes a part of path with the distance of one unit. If there is no transfer, the travel distance of p_1 is $d(p_1) = 9$, and travel distance of p_2 is $d(p_2) = 5$. If a cluster of requests (three requests) can transfer, the travel distance of p_1 is $d'(p_1) = 5$ and the travel distance of p_2 is $d'(p_2) = 6$. $d'(p_1) + d'(p_2) = 11 < d(p_1) + d(p_2) = 14$. The total travel distance of two PVs is reduced by 3.

6 Performance evaluation

Our simulation is built under Windows OS based on the road network of Shanghai in China, particularly, in the downtown area of about 50 km^2. Little work is conducted on the area of traffic speed prediction [23], and it is not the focus of this paper. We use the average speed of PVs for simplicity. We assume that the earliest start time is 0. We use the requests of Shanghai taxis data in this area.

Table 3 lists the values of variables in our simulations. We put 50 PVs to the system, and then according to the range of α (1 to 5) generate 50 to 250 requests/passengers. We first calculate the paths of all PVs using the no transfer solution in [1], and then optimize the paths using the proposed algorithms. The following are the results of the simulations.

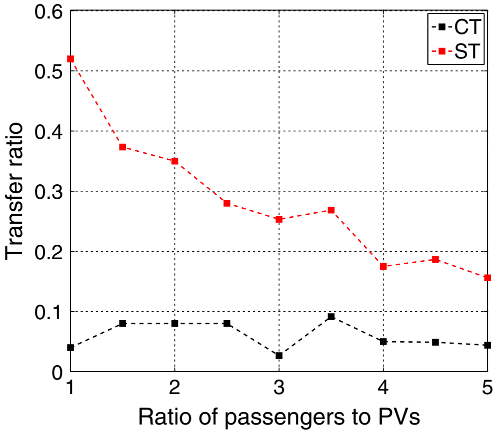


Fig. 5 Transfer ratio with different α

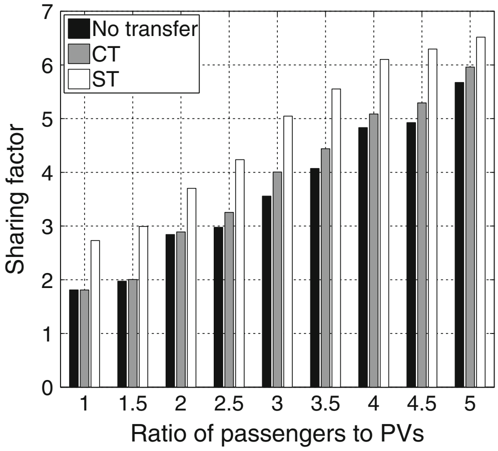


Fig. 6 Sharing factor with different α

Figure 5 presents the ratio of transferred passengers. We know that, more passengers may transfer using ST compared with CT. The transfer ratio using ST decreases with the increasing of α. However, the transfer ratio using CT almost does not change. Generally, the transfer ratio does not exceed 10 % using CT. If $\alpha \leq 2$, the transfer ratio using ST is higher than 25 %, which is much higher than that using CT. The reason is that, the transfer limits using CT are more than that using ST: in CT some requests are clustered as one. CT may have to search one transfer point for several passengers with QoS guarantee so that they transfer at the same location.

The average traffic sharing factor is denoted by h, which can be calculated using Eq. 3. Figure 6 depicts the sharing factor of passengers, which implies the condition of traffic sharing. We see that, the sharing factor using ST is larger than that using CT or no transfer solution. The sharing factor will increase with the growing of α. If more passengers

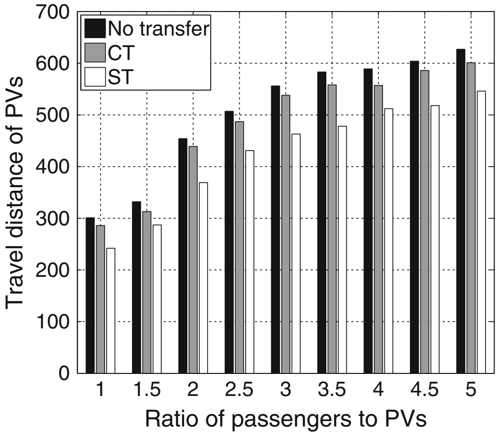


Fig. 7 Travel distance with different α

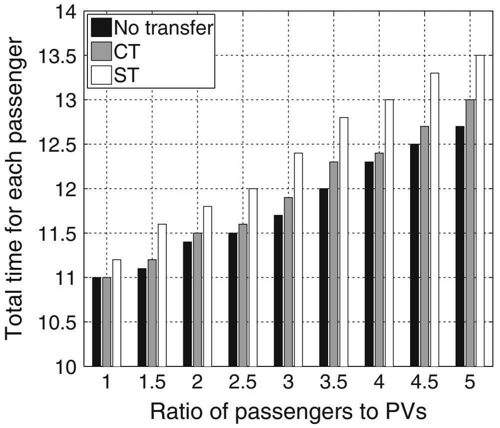


Fig. 8 Total time for each passenger with different α

enter the cloud-based PV system, more passengers share one vehicle no matter using ST or CT.

$$h = \sum_{r \in R} d(r) / \sum_{p \in P} d(p) \tag{3}$$

From Fig. 7, we see that, the travel distance of PVs grows if more passengers need trip service using ST or CT, and this can be predicted by all of us. The total travel distance of PVs is reduced by 15 and 4 % respectively using ST and CT, compared with the no transfer solution. In Fig. 8, we see that, the total time for each passenger including waiting time, transfer time (if transfer) and traveling time is almost the same under three solutions. The total time for each passenger using ST may be only a little longer than that using no transfer solution or CT. Generally, the total time increases with more passengers enter this system no matter using ST or CT.

The experiments show that, the average transfer time using ST is 2.4 minutes, which is a little longer than that using CT (2.0 minutes). We conclude that, ST or CT has little effect on the total trip time of passengers, due to that we have considered the transfer time and detour ratio in transfer solutions. All the passengers can enjoy their trips with sustainable comfort such as short transfer time, low detour ratio.

7 Discussion and conclusion

The PV system is a cloud-assisted ridesharing system with several advantages, e.g., low-cost, high traffic resource sharing, high efficiency, and low carbon emission. To further improve the traffic sharing and efficiency, we study the transfer problem in the PV system, which is NP-Complete. We propose two transfer strategies, ST and CT under sustainable discomfort (denoted by QoS constraints) based on the previous work. The simulations based on the Shanghai road network have analyzed the performance of propose algorithms. Generally, the performance of ST is better CT. Meanwhile, CT can also improve the transfer efficiency with sacrificing certain performance. The total travel distance of PVs is reduced using ST and CT compared with the no transfer solution.

The solutions we have proposed can also be used in the current ridesharing systems such as Uber Pool, Lyft, although they are not as dynamic as the PV system. In future, we would consider the transfers between PVs and subways, taxis, or buses in smart cities. The whole traffic efficiency would be improved with the corporation of several transportation systems. Some future work consists of the planning for transfers in dynamic settings, such as personal preference and interests.

Acknowledgments This work was supported by the Natural Science Foundation of China (NSFC) projects (Nos. 61373155, 91438121 and 61373156).

References

1. Zhu M, Liu X-Y, Tang F, Qiu M, Shen R, Shu W, Wu M-Y (2016) Public vehicles for future urban transportation systems. (under revision) IEEE Transactions on Intelligent Transportation Systems (TITS)
2. Zhou Z, Zhao D, Xu X, Du C, Sun H (2015) Periodic query optimization leveraging popularity-based caching in wireless sensor networks for industrial iot applications. Springer Mobile Networks and Applications (MONET) 20(2):124–136
3. Drews F, Luxen D (2013) Multi-hop ride sharing. In: Sixth Annual Symposium on Combinatorial Search
4. Teubner T, Flath CM (2015) The economics of multi-hop ride sharing. In: Business & Information Systems Engineering, pp 1–14
5. Galland S, Knapen L, Yasar A.-U.-H., Gaud N, Janssens D, Lamotte O, Koukam A, Wets G (2014) Multi-agent simulation of individual mobility behavior in carpooling. In: Transportation Research Part C: Emerging Technologies
6. Handke V, Jonuschat H (2013) Flexible ridesharing. Springer
7. Qiu M, Zhong M, Li J, Gai K, Zong Z (2015) Phase-change memory optimization for green cloud with genetic algorithm. IEEE Trans Comput (TC) 64(12):3528–3540
8. Wan J, Zhang D, Sun Y, Lin K, Zou C, Cai H (2014) Vcmia: a novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing. Springer Mobile Networks and Applications (MONET) 19(2):153–160
9. Qiu M, Chen Z, Ming Z, Qin X, Niu J (2014) Energy-aware data allocation with hybrid memory for mobile cloud systems. IEEE Syst J PP(99):1–10
10. Li Y, Dai W, Ming Z, Qiu M (2015) Privacy protection for preventing data over-collection in smart city. IEEE Trans Comput (TC) PP(99):1–14
11. Bit-Monnot A, Artigues C, Huguet M-J, Killijian M-O (2013) Carpooling: the 2 synchronization points shortest paths problem. In: OASIcs-OpenAccess Series in Informatics, vol 33, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik

12. Trasarti R, Pinelli F, Nanni M, Giannotti F (2011) Mining mobility user profiles for car pooling. In: ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp 1190–1198
13. Zhang D, Li Y, Zhang F, Lu M, Liu Y, He T (2013) Coride: carpool service with a win-win fare model for large-scale taxicab networks. In: ACM Conference on Embedded Networked Sensor Systems (SenSys)
14. Coltin B, Veloso M (2014) Ridesharing with passenger transfers. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3278–3283
15. Ray J-B (2014) Planning a real-time ridesharing: network Critical mass and role of transfers. In: Transport Research Arena (TRA) Conference: Transport Solutions from Research to Deployment
16. Ben Cheikh S, Hammadi S, Tahon C (2014) Based-agent distributed architecture to manage the dynamic multi-hop ridesharing system. In: IEEE International Symposium on Network Computing and Applications (NCA), pp 101–104
17. Bouros P, Sacharidis D, Dalamagas T, Sellis T (2011) Dynamic pickup and delivery with transfers. In: Springer Advances in Spatial and Temporal Databases, pp 112–129
18. Masson R, Lehuédé F, Péton O (2012) Simple temporal problems in route scheduling for the dial–a–ride problem with transfers. In: Springer Integration of AI and OR Techniques in Contraint Programming for Combinatorial Optimzation Problems, pp 275–291
19. Zhu M, Liu X-Y, Kong L, Shen R, Shu W, Wu M-Y (2014) The charging-scheduling problem for electric vehicle networks. In: IEEE Wireless Communications and Networking Conference (WCNC), pp 3178–3183
20. Hong kong: Airport express, bus or taxi? http://www.tripadvisor.com/Travel-g294217-c178763/Hong-Kong:China:Airport.Express.Bus.Or.Taxi.html
21. Uber proves 40 % cheaper than sydney taxis and more reliable in choice test. http://www.theguardian.com/australia-news/2015/sep/24/uber-cheaper-sydney-taxis-more-reliable-choice-test
22. Hou Y, Li X, Qiao C (2012) Tictac: From transfer-incapable carpooling to transfer-allowed carpooling. In: IEEE Global Communications Conference (GLOBECOM), pp 268–273
23. Asif MT, Dauwels J, Goh CY, Oran A, Fathi E, Xu M, Dhanya MM, Mitrovic N, Jaillet P (2014) Spatiotemporal patterns in large-scale traffic speed prediction. IEEE Trans Intell Transp Syst (TITS) 15(2):794–804