TicTac: From Transfer-Incapable Carpooling to Transfer-Allowed Carpooling

Yunfei Hou, Xu Li, *IEEE Member*, and Chunming Qiao, *IEEE Fellow*
Department of CSE, State University of New York at Buffalo, USA

Abstract-Current *transfer-incapable carpooling* (*TIC*) scheme cannot fully utilize vehicles' available space because a carpooling passenger has to go from her origin to her destination by getting a ride from only one vehicle. This is akin to insist on delivering some packets only using one-hop communications, which usually performs worse than allowing multi-hop communications. In this paper, inspired by the "Store-and-Forward" strategy used in Delay-Tolerant Networks (DTN), we propose a new carpooling paradigm called *transfer-allowed carpooling (TAC)*, with which each passenger can be served by more than one vehicle to go from her origin to her destination, thus increasing the carpooling performance. In particular, when given *a*) a number of carpooling requests (each with a maximum waiting-time and a maximum number of transfers for a passenger), and *b*) a list of participating vehicles (each specifying a maximum detour distance for a driver), we address a new optimization problem called *Transfer-Allowed Carpooling* whose objective is to maximize the *successful carpooling ratio* (*SCR*). Two effective strategies have been proposed from a driver and passenger standpoint, respectively. In addition to conducting large-scale simulations, we also present a case study in a more realistic setting by utilizing real routes collected from taxis in the city of Shanghai. Our major results are: 1) the proposed TAC approach can significantly improve SCR (by 35% to 60%), compared to the traditional TIC approach; and 2) allowing one transfer (i.e., the maximum number of transfers=1) improves the carpooling efficiency most, while allowing more than one transfer does not bring any noticeable benefits.

Keywords-Intelligent Transportation Systems; Delay-Tolerant Networks; Transfer-Allowed Carpooling; Rideshare Planning; Real Case Study;

I. INTRODUCTION

Transportation is the backbone of smart growth and economic development. Urban traffic congestion and air pollution are two serious problems for cities today. We need
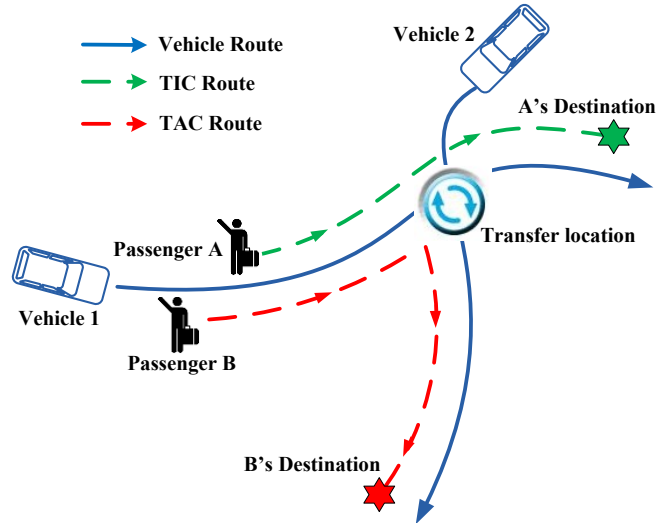
Fig.1 The examples illustrating Transfer-Incapable Carpooling (TIC) and Transfer-Allowed Carpooling (TAC)

forwarding-thinking transportation method that encourage mass transit, promote green routing, and advance transportation systems that give communities transportation options and reduce traffic congestions.

Prior research has shown that carpooling is a promising approach for reducing road traffic as well as CO_2 emissions [1]-[7]. However, the existing solutions to rideshare planning cannot fully utilize vehicles with available space. In particular, we have the following observation:

Firstly, traditional carpooling systems usually operate in a *transfer-incapable carpooling* (*TIC*) mode. In other words, a carpooling passenger is only considered to be served by one vehicle. Accordingly, such a system will fail to provide a feasible rideshare plan to a passenger if she cannot be taken to her destination by one vehicle. As shown in Fig. 1, two passengers *A* and *B* need carpooling services. With TIC, passenger *A* can take vehicle 1 to her destination while passenger *B* cannot be served because no single vehicle can both pick up/drop off her at her source/destination, respectively. Actually, such one-hop delivery model cannot yield as a good performance as multi-hop communications either in telecommunications networks. Therefore, this motivates us to borrow the concept of the "Store-and-

Forward" strategy used in Delay-Tolerant Networks (DTN) [9] to carpooling application. In particular, we propose to allow a passenger to transfer from one vehicle to another at an intermediate location such that multiple vehicles can cooperatively serve one carpooling request (In the rest of the paper, request/passenger will be used interchangeably).

It is worth noting that routing protocols in DTN cannot be applied to rideshare planning. This is because DTN routing normally operates at each individual node by defining node behaviors such as buffer management, neighbor selection, etc, whereas a rideshare plan has to be determined at a global/network level in order to systematically plan routes for vehicles, and let them serve a set of requests cooperatively. In addition, DTN routing considers data delivery issues for given node mobility patterns, while in our carpooling application, vehicle mobility/routes could be modified in order to pick up and drop off passengers.

In this paper, we study a new carpooling paradigm called *transfer-allowed carpooling (TAC)*. Referring to Fig.1 again, passenger *B* can be served at this time if transfer is allowed, i.e., she can be picked up by vehicle 1 at her source and then transfer to vehicle 2 at a transfer location, and finally be dropped off at her destination by vehicle 2. In particular, when given *a)* a number of passengers (each with a maximum waiting-time and a maximum number of transfers for a passenger), and *b)* a list of participant vehicles (each specifying a maximum detour distance for a driver), we address a new optimization problem called *Transfer-Allowed Carpooling* whose objective is to maximize the *successful carpooling ratio* (*SCR*), which is the ratio of successfully served requests to the total number of carpooling requests.

The major contributions of this paper are as follows:

1. To the best of our knowledge, no existing work has looked into the TAC paradigm. In particular, our major findings are: *a)* TAC can significantly improve SCR compared to TIC (by 35% to 60% both in our simulations and the real case study); and *b)* Allowing one transfer per passenger improves the carpooling efficiency most, while allowing more than one transfer does not bring any noticeable benefits.

2. From both a driver and passenger's standpoint, we introduce two effective rideshare planning strategies to TAC, i.e., *Driver Experience-aware Strategy* (*DES*) and *Passenger Experience-aware Strategy* (*PES*), and perform large-scale simulations.

3. We also present a case study in the city of Shanghai by using real routes collected from taxis, and provide useful insights.

The rest of the paper is organized as follows: in Sec. II, we formally describe the TAC problem. We propose two effective strategies in Sec. III and report the results from simulations in Sec. IV. A case study in the city of Shanghai is presented in Sec. V. We discuss the related work in Sec. VI, and Sec. VII concludes the paper.

II.   THE TRANSFER-ALLOWED CARPOOLING (TAC) PROBLEM

In this section, we define the proposed TAC problem.

We consider rideshare planning by a management center, which has the information about the current planned routes of *m* participant vehicles offering rides and *n* carpooling requests/passengers that need to be served. As mentioned earlier, we consider the TAC mode, in which it is possible for a passenger to transfer from one vehicle to another, and eventually they will be taken to their destinations. A request will fail if there is no feasible rideshare plan that can take the passenger to her destination. We aim to design an optimal rideshare plan, with which the number of successfully served requests can be maximized.

A.   Carpooling Request Model

Let $R=\{r_i\}$, $i=1, 2\ldots n$ be the set of n carpooling requests. Typically, a request r_i can be defined as a 5-tuple {src_i, $dest_i$, $t\_latest\_depart_i$, MAX_TRANS_i, MAX_WAIT_i}, the first three parameters are the origin location, the destination location and the latest departure time, respectively. Note that we assume that a passenger can be picked up at any time before $t\_latest\_depart_i$, i.e., if her rideshare request is granted, she can make herself ready at her origin at any time earlier than $t\_latest\_depart_i$ and she does not mind waiting till $t\_latest\_depart_i$ before being picked up. However, from a passenger perspective, two other factors can affect her trip experience: the number of transfers and the waiting time during transfers. Accordingly, we use MAX_TRANS_i and MAX_WAIT_i to denote the maximum number of transfers and the maximum/tolerable waiting time during a transfer for each request r_i (Note that, we assume that vehicles will not wait for passengers, therefore, during a transfer, a passenger also needs to arrive at the pre-determined transfer location before the arrival of the next vehicle that will pick her up). In addition, it is easy to see that the traditional TIC is the special case of the current TAC model if we set $MAX\_TRANS_i=0$.

B.   Carpooling Service Model

Let $V=\{v_j\}$, $j=1,2\ldots m$ be the set of *m* participant vehicles. Typically, a vehicle will send a carpooling service information to the rideshare management center, which can also be defined as a 5-tuple {src_j, $dest_j$, $planned\_route_j$, MAX_DETOUR_j, C_j}, the first three parameters are vehicle's the origin location, the destination location and the current planned route, respectively. C_j is the seat capacity of v_j and MAX_DETOUR_j is the maximum detour length for v_j. Note that, to some extent, MAX_DETOUR_j also limits the delay time of drivers due to their carpooling activities. After a vehicle accepts a carpooling request, then $planned\_route_j$ will also be updated accordingly.

Let $R(j) \subseteq R$ be the set of passengers already in vehicle v_j (initially, $R(j) = \varnothing$ and $|R(j)| \leq C_j$). We denote by $ORIGIN\_TRIP\_LENGTH_j$ the length of $planned\_route_j$ when $R(j)=\varnothing$. Normally the original route follows the

shortest path from driver's source to her destination. $R(j) \neq \varnothing$ means that v_j are involved in carpooling activities. Since each passenger needs to be picked up and dropped off by the carpooling vehicles at their own sources and destinations, we denote by $CARPOOLING\_TRIP\_LENGTH_j$ the actual travel distance from v_j's source to destination with carpooling activities. Then, a feasible rideshare plan needs to satisfy the following constraint:

$$MAX\_DETOUR_j \geq CARPOOLING\_TRIP\_LENGTH_j - ORIGIN\_TRIP\_LENGTH_j \qquad (1)$$

C. Problem Statement

Now, we give a formal description of TAC and the performance metrics we use to evaluate the solutions.

Definition 1 *The Transfer-Allowed Carpooling (TAC) Problem*: Given a set of requests $\{r_i\}$ and vehicles $\{v_j\}$, decide new routes for vehicles (in which we allow passengers to transfer between multiple vehicles) so that the number of passengers that could be transited to their destinations can be maximized, by considering the constrains on *MAX_TRANS* and *MAX_WAIT* for passengers and *MAX_DETOUR* and seat capacities for vehicles.

Definition 2 With a given solution to TAC, we denote the set of requests that can be successfully served by R', we define the *Successful Carpool Ratio* (*SCR*) of this solution to be:

$$SCR = \frac{R'}{|R|} \qquad (2)$$

III. STRATEGIES DESIGN FOR TAC

In this section, we design two solution strategies for TAC, which look into the problem from a vehicle and passenger standpoint, respectively. Due to lack of space, we will outline the algorithmic steps but omitting other details.

A. Driver Experience-aware Strategy (DES)

Since the incentive for carpooling is not only to improve traffic efficiency but also to reduce gas consumption and CO_2 emissions (which are largely related to vehicle travel distance). It is natural for us to design rideshare plans with distance-related concerns. Typically, *Driver Experience-aware Strategy* (*DES*) processes one request at a time. When processing each request, it first finds all feasible rideshare plans for the current request and selects the one with minimum additional detour due to picking up and dropping off the related passenger. After processing a request, the routes of vehicles will be updated, which will then be used to serve other carpooling requests in a later time. In addition, we assume that all vehicles/passengers will accept/follow any rideshare plan as long as it satisfy their constraints on *MAX_DETOUR* and *MAX_WAIT*. Typically, DES works as follows:

Step 1) We refer this step as *Request Ranking and Selection Process*. Conceptually, we first build a bipartite graph in order to decide which request should be processed.

In the bipartite graph, we have request on one side, and all vehicles that can possibly pick up (without considering transfer/destination) on the other, then we order the requests accordingly. The request that has the fewest eligible vehicles is served first. Here, we use this approach to *roughly estimate* the probability of whether a request can be served by some vehicles. The basic idea is that the less vehicles that can pick up the passenger, the higher priority we should give to the according request, because such a request will hardly get a chance to be served if other requests have already taken many carpooling resources.

Step 2) We refer this step as *Actual Vehicle Selection Process*. This step is to calculate all feasible rideshare plans for the request selected to be processed in Step 1). For a given request, we calculate feasible rideshare plans by considering four major aspects: *a*) whether each of the vehicles involved in a proposed rideshare plan could serve the passenger without violating their *MAX_DETOUR*; *b*) whether at each transfer location, the arrival time of the passenger is earlier than the arrival of the next carpooling vehicle so that she will not miss the next ride; *c*) whether the waiting time during each transfer is less than *MAX_WAIT*; and *d*) whether all the vehicles in a rideshare plan still have available seat to serve this request. With *MAX_TRANS*, this step will produce all the feasible rideshare plans for the current request.

Step 3) For all possible rideshare plans for a given request, we select the one with minimum additional detour induced by serving this request.

Step 4) Once a request has been processed, we update vehicle status and routes, then go back to Step 1). The procedure will terminate when no more request can be served.

B. Passenger Experience-aware Strategy (PES)

The number of transfer and the waiting time during a transfer are also important factors that affect passengers' travel experience. Addressing the problem from a passenger standpoint, *Passenger Experience-aware Strategy* (*PES*) aims to provide rideshare plans not only with a fewer transfers, but also with less waiting time. In addition, unlike in DES which examines one request at a time, PES will deal with a batch of requests in each round. In particular, for each batch, we will try to satisfy as many requests with zero transfer first, and then as many requests with one transfer next, so on so forth. Note that, the actual vehicle selection process used to in DES will be reused in PES. Typically, PES has the following procedures:

Step 1) In order to reduce transfer time, in each round, we find rideshare plans for requests by limiting the number of transfer by a temporary threshold, which is denoted by $N_{transfer} < MAX\_TRANS$. Therefore, we start with finding rideshare plans with no transfer (i.e., $N_{transfer} = 0$) for each request based on vehicle's current routes and status. In this step, multiple possible rideshare plans might be available for one request. At the same time, one vehicle might also be involved in multiple rideshare plans or different requests

Table I: Default values of experiment parameters

No. of Requests (n)	30
No. of Vehicles (m)	12
No. of Available seat in Each Vehicle (C)	4
Maximum Tolerable Waiting Time (MAX_WAIT)	5 min
Maximum Number of Transfer (MAX_TRANS)	1
Maximum Tolerable Detour (MAX_DETOUR)	5 miles
The Length of a Road Section	1~3 miles
Avg. Travel Speed on a Road Section	20-40 mph
Minimum Travel Distance of A Request	5 miles

(which may cause a conflict because PES will separately check the requests).

Step 2) We focus on the requests which have been provided with one or more rideshare plans from Step 1). For each of such requests, we select the rideshare plan having the earliest arrival time so that passengers' waiting time and total trip time can be reduced. After processing a request, we also need to update vehicles' routes and status before we check the next request. In particular, if there is a conflict between two requests, we simply skip this request and reconsider it in the next round. When all requests have been examined, go to step 3).

Step 3) We increase $N_{transfer}$ by one (i.e., $N_{transfer}$= $N_{transfer}$+1) and go back to Step 1) for next round. This will search for carpooling opportunities with more transfers. The procedure exits when all the requests are handled or MAX_TRANS is reached.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed solutions for TAC. In our simulation, we focus on a road network in downtown area (8×8 grid), in which vehicles travel through the area and requests are generated in a random manner. Table I shows the default values of the parameters in our simulation.

Fig. 2 (a) shows the performance of DES and PES respectively in the default setting. It also shows SCR using the traditional TIC approach (Note that since TIC is a special case of TAC as mentioned earlier, we can directly apply DES by setting MAX_TRANS=0. Actually, both DES and PES will have the same procedure when MAX_TRANS=0). Generally speaking, TAC can significantly increase carpooling performance in terms of higher SCRs, compared to TIC. DES performs slightly better than PES because with DES, each vehicle opts to spend as little detour as possible to serve one request, and therefore can save more detour budget for other passengers; with PES, the waiting time is the major consideration, however, saving the waiting time for a single passenger cannot increase carpooling opportunities for others.

Next, we are interested in how MAX_TRANS can affect carpooling performance. In particular, we pay more attention to DES since it has the best performance in the default setting. Fig. 2(b) shows, once again, that TAC with

one transfer can improve carpooling performance by almost 50% over TIC; however, it also shows that increasing MAX_TRANS cannot further bring much benefit in regards to SCR. For example, DES almost produces the same performance when $MAX\_TRANS \geq 2$. It is worth noting that such a finding has two practical implications: 1) In reality, we may pay more attention to one-transfer rideshare planning, which could significantly reduce computational complexity; and 2) One-transfer rideshare avoids making an excessive number of transfers and results in a better passenger experience.

Fig. 2 (c) compares SCRs of the two solutions as the number of request increases. Clearly, when more passengers are involved, only a small portion of requests can be successfully served because of limited carpooling resources, e.g., detour length, available seats, etc, and therefore this leads to a decreased SRC.

In Fig. 2 (d) and (e), we examine how two other factors can affect SCR, i.e., the maximum detour for vehicles (MAX_DETOUR) and the maximum waiting time for passengers (MAX_WAIT). We observe that increasing MAX_DETOUR can improve SCR for both DES and PES, which can be explained as follows: when more detour budget is available, vehicles can afford longer detour to pick up / drop off passengers, thus increase carpooling opportunity, which is shown in Fig. 2 (d). Fig. 2 (e) shows that increasing MAX_WAIT can also improve SCR. This implies that a patient passenger has a better chance to find a ride. However, after a certain point (e.g., 5 minutes in our simulation), increasing MAX_WAIT will not further improve SCR. These two figures imply that whether a passenger can be served largely depends on vehicles' detour budget.

Lastly, we investigate other practical impacts of TAC. Fig. 2 (f) shows the total travel distance of all the participant vehicles with different strategies. We assume that if a request cannot be served, the passengers will either drive their own cars or take taxis to their destinations. Their travel distance thus should also be included into the total travel distance. We have the following observations: 1) Carpooling can reduce the total travel distance compared to no-carpooling case; 2) Compared to TIC, TAC can further reduce the total travel distance, which is another benefit of TAC in terms of gas saving and emission reduction.

V. A CASE STUDY IN SHANGHAI

In this section, we present a case study in the city of Shanghai using real routes collected from 4000 taxis. In this simulation, we focus on a 7KM × 6KM area (as shown in Fig. 3) and intend to use the real routes travelled by 12 taxis within this area. Note that we have the following technical challenge: on one hand, we want to perform our testing on these real routes without artificially modifying them, but on the other hand, the taxis from the data did not perform carpooling, while each vehicle in our carpooling study may
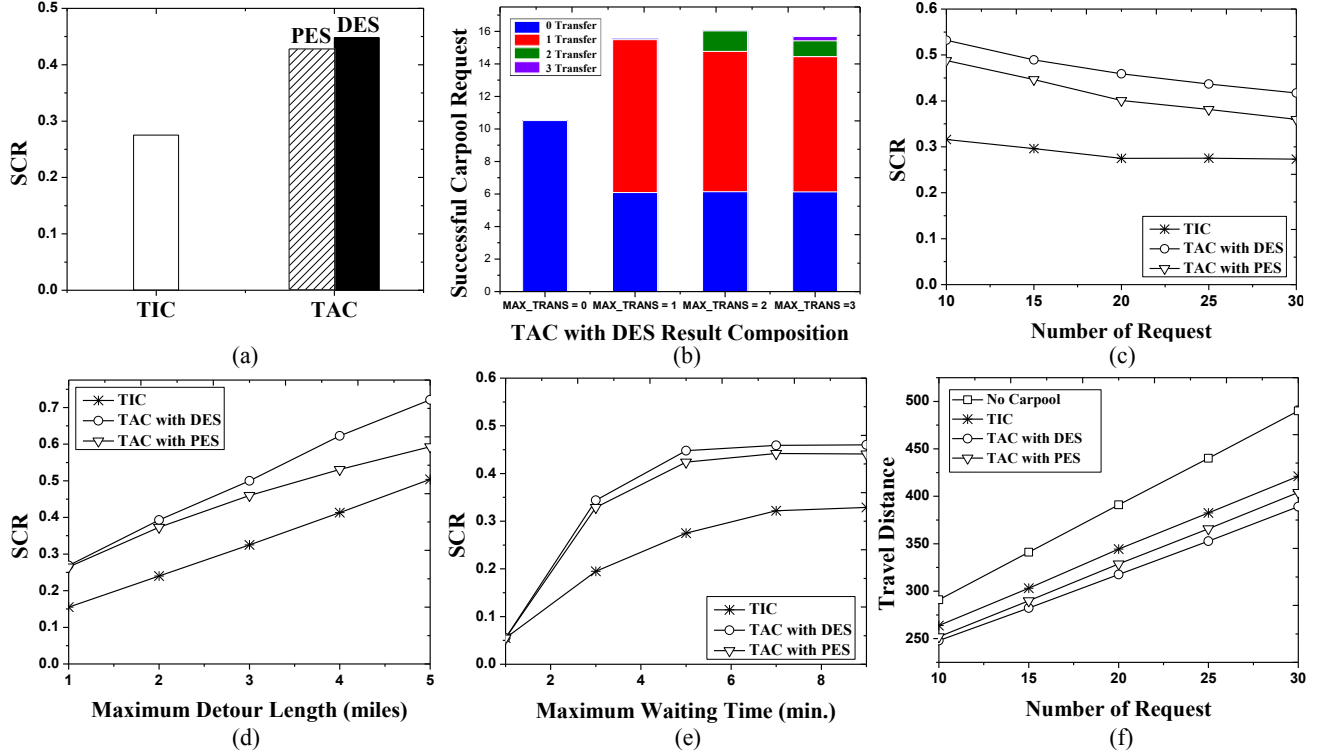
Fig. 2. Testing results of TAC/TIC. (a) Performances with default setting. (b) Result Composition of DES. (c) Performances with different request number. (d) Performances with different *MAX_DETOUR*. (e) Performances with different *MAX_WAIT*. (f) Total travel distance with different request number.

change their routes from its original plan, i.e., they can dynamically change their routes to pick up and drop off passengers. Therefore, in this section, we consider a special case of TAC, in which we do not allow vehicles to take any detour from their original routes, i.e., *MAX_DETOUR*=0 so that the real route data can be used with minimum modification. We build our testing scenario as follows: we first plot the entire routes of vehicles and identify the intersections of these routes. Then, we use these intersections and links (a segment between two intersections) to build a virtual road network (The links' average speed is set as the vehicles' travel speed in the data). From the route data, we also find that many vehicles travel to similar destination areas, which are either transportation center or business districts. Therefore, we use the intersections in those areas as the destinations of requests, while the sources of the requests are randomly generated in the road network. As the requests' sources/destinations are selected from the intersections of the routes of vehicles, a request could still be served by TAC rideshare plan even if *MAX_DETOUR*=0.

Fig. 4 (a) compares SCRs of the two solution strategies as the number of request increases. It again confirms that allowing transfers can increase carpooling performance and SCRs decrease with the increased number of requests. DES and PES yield the same performance at this time because the major difference between DES and PES lies in how to use

the detour budget of vehicles, but in this case study we set *MAX_DETOUR*=0.

In Fig. 4 (b), we examine the effect of increasing the maximum waiting time on SCR in this case study. Compared to Fig. 2 (e), we can also see that 1) increasing *MAX_WAIT* is beneficial for SCR, because an impatient passenger with a small *MAX_WAIT* can hardly be satisfied due to the tight transfer schedule; 2) After some threshold value (e.g., 12 min in the case study), increasing *MAX_WAIT* cannot make further improvement because no matter how long a passenger waits for a transfer, there is no vehicle passing around her to serve her carpooling request. Overall, the results from this case study (based realistic route data) also suggest that TAC can achieve 32.3%-36.4% improvement over traditional TIC.

VI. RELATED WORK

In this section, we review the existing works on carpooling. Overall, most of existing works looked into the carpooling problem by assuming that one request can only be served by one vehicle while our work introduce the concept of transfer-allowed carpooling paradigm to improve carpooling performance.

In [1], the authors proposed Vehicle-to-Passenger (V2P)

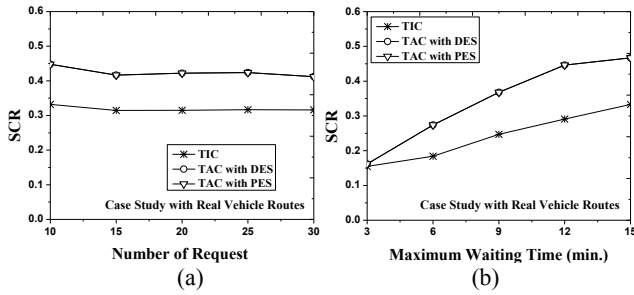Fig.3 The scenario and road network in the case study

Fig. 4. Testing results in the case study. (a) Performances with different request number. (b) Performances with different *MAX_WAIT*.

communication, which allows direct, instant, and flexible communication between moving vehicles and roadside passengers so that the passengers can take a free ride or call a taxi via radio queries over VANETs. Work in [5] developed and tested computational methods for guiding collaboration that demonstrate how shared plans can be created in real-world settings. [2] tried to handle carpooling request by constructing distributed dynamic graph. They decomposed the information about passengers and driver into subgroups, so that the carpooling requests can be handled in parallel process. However, the implementation issue and performance validation have not been reported in this paper. [3] studied taxipooling problem and proposed two greedy or time-space graph based algorithms for sharing a taxi to/from a same destination, i.e., one origin to many destinations or many origins to one destination, respectively. Work in [6] focused on on-line car pooling service to facilitate matching of drivers and riders . In particular, the authors have developed a smart ride-share system with an efficient scheduling algorithm for ride sharing, which can potentially achieve better vehicle utilization, energy consumption and user convenience. In [7], the authors proposed a dynamic taxi-sharing system, which can immediately serve each irregular ride-sharing request and find a fuel-saving taxi for it.

In addition, several existing studies in the VANET field have focused on other interesting applications, including on-road service delivery [8], data access [9], traffic signal schedule prediction [11], driver safety [12], etc.

VII. CONCLUSION

In this paper, we introduced a new carpooling paradigm called *transfer-allowed carpooling*, which aims to increase carpooling performance by trying to fully utilize vehicles' available space. In particular, we addressed a new *Transfer-Allowed Carpooling* (*TAC*) *Problem* with the objective of maximizing *successful carpooling ratio* (*SCR*) for a given number of passengers and carpooling vehicles. We proposed *Driver Experience-aware Strategy* (*DES*) *and Passenger Experience-aware Strategy* (*PES*) for rideshare planning, and performed large-scale evaluation. We also presented a case study in the city of Shanghai by using real route data collected from taxis. From this work, our major finding is that on one hand, TAC can significantly improve SCR (by 35% to 60%), compared to the *transfer-incapable carpooling* (*TIC*); on the other hand, allowing one transfer is the most effective option while allowing more than one transfer does not bring us any noticeable benefits.

REFERENCES

[1] N. Liu, M. Liu et al "When Transportation Meets Communication: V2P over VANETs," *IEEE ICDCS*, 2010.

[2] M, Sghaier, H. Zgaya, "A Novel Approach Based on a Distributed Dynamic Graph Modeling Set Up over a Subdivision Process to Deal With Distributed Optimized Real Time Carpooling Requests," *IEEE ITS*, 2011.

[3] C. C. Tao and C. Y. Chen, "Heuristic Algorithms for the Dynamic Taxipooling Problem Based on Intelligent Transportation System Technologies", *IEEE FSKD*, 2007.

[4] J.-F. Cordeau, G. Laporte, The Dial-a-Ride Problem: Models and Algorithms," *Annals of Operations Research*, 2007.

[5] E. Kamar and E. Horvitz, "Collaboration and Shared Plans in the Open World: Studies of Ridesharing *IEEE IJCAI*, 2009.

[6] C. -M. Chen; D. Shallcross, "Smart Ride Share with Flexible Route Matching," *IEEE ICACT*, 2011.

[7] P. -Y. Chen et al "A Fuel-Saving and Pollution-Reducing Dynamic Taxi-Sharing Protocol in VANETs," *IEEE VTC*, 2010 Fall.

[8] X. Li, X. Yu, A. Wagh, and C. Qiao, "Human Factors-aware Service Scheduling in Vehicular Cyber-Physical Systems," *INFOCOM*, 2011.

[9] S. Jain, K. Fall et al" Routing in a Delay Tolerant Network," *ACM SIGCOMM*, 2004.

[10] X. Hou, P. Deshpande and S. R. Das, "Moving Bits from 3G to Metro-scale WiFi for Vehicular Network Access: An Integrated Transport Layer Solution," *IEEE ICNP*, 2011.

[11] E. Koukoumidis, L. Peh, et al "SignalGuru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory," *ACM MOBISYS*, 2011.

[12] J. Yang, "Detecting Driver Phone Use Leveraging Car Speakers," *ACM MOBICOM*, 2011.