ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 6, June 2016

Real Time Taxi Ride Sharing System

Shankar N[1], Venkatesh K[2]
M.E Computer Science Engineering[1], Assistant Professor/ Computer Science Engineering[2]
Mount Zion College of Engineering and Technology, Pudukottai[1, 2]

Abstract Taxi sharing system receives passenger's real time request sent from smartphones and schedules proper taxis to pick up by taxi sharing with respect to passenger capacity, time and monetary constraints. The monetary constraint provides benefit for both passenger and taxi drivers: passenger will not pay more compared with no ridesharing and drivers get more profit compared with no ridesharing. While this system is of significant social and environmental benefits e.g., reducing traffic, saving fuel consumption and satisfying people's commute, real time taxi pooling is not well studied yet. To this end, we improved the taxi pooling system by using mobile-cloud architecture. Taxi drivers and taxi riders use the taxi pooling service by smartphones. Initially, cloud gets passenger's ride request and finds appropriate taxi for the customer by using taxi searching algorithm supported by spatio-temporal index. A scheduling process is then performed in the cloud to select a taxi that satisfies the request with minimum increase in travel distance. When the ratio of the number of ride requests to the number of taxis is 6, our proposed system serves three times as many taxi riders as that when no ridesharing is performed while saving 11 percent in total travel distance and 7 percent taxi fare per rider.

Keywords*:* Taxi sharing, Spatial database and Intelligent transportation system

I. INTRODUCTION

Taxi is an important transportation mode between public and private transportations, delivering millions of passengers to different locations in urban areas. However, taxi demands are usually much higher than the number of taxis in peak hours of major cities, resulting in that many people spend a long time on roadsides before getting a taxi. Increasing the number of taxis seems an obvious solution. But it brings some negative effects, e.g., causing additional traffic on the road surface and more energy consumption, and decreasing taxi driver's income (considering that demands of taxis would be lower than number of taxis during off-peak hours).

In our country, air pollution is the major problem. In Delhi, air pollution is measured by 600 µg/m³. But average air content should be 60(µg/m³) only. To solve this issue, Delhi government announced new Odd Even Vehicle Rule. This rule would define which car you can drive on a particular date. On even dates, only cars with license plates ending with an even number will be allowed on city roads, and vice versa. Our project will reduce the air pollution because of sharing taxi among people [1].

The fuel energy consumption in India is the fourth biggest after China, USA and Russia. India imports nearly 75% of its 4.3 million barrels per day crude oil needs. The net import of crude oil & petroleum products is 146.70 million tons worth of Rs.5611.40 billion. It cause more expenditure to the country [2]. This system will reduce the fuel consumption of the commercial taxis with the rate of minimum 10%. According to Transportation Ministry, 17 lakhs Commercial Taxis are newly registered with in a year.

India is the most traffic congestion country in the world because of population of our country. Traffic is the big problem of the country. In cities, 10kmph is the average travelling speed of vehicles. It causes fuel wastage. The study covered 17 major routes and found that national average of fuel mileage is only 3.96 km per litre. While it's maximum on Chennai-Kolkata corridor, it's the least at 3.44 km on the Delhi-Nagpur stretch. The country loses Rs 60,000 crore a year due to congestion (including fuel wastage), slow speed of freight vehicles and waiting time at toll plazas and checking points, a study on operational efficiencies of freight transportation by roads has claimed [3].

This project benefits of reducing traffic, fuel consumption and air pollution in cities significantly. Moreover it increases the taxi usage to ground level people because of economic benefits.

II. THE REAL-TIME TAXI-SHARING PROBLEM

The real-time taxi-sharing problem consists of a data model, constraints, and an objective function. We describe each part separately below before giving the formal definition of the problem.

A. Ride Request:

A ride request R is associated with a timestamp R.t indicating when R was submitted, a origin point R.o, a destination point R.d, a time window R.pw defining the time interval when the rider wants to be picked up at the origin point, and a time window R.dw defining the time interval

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 6, June 2016

when the rider wants to be dropped off at the destination point.

The early and late bounds of the pickup window are denoted by R.pw.e and R.pw.l respectively. Likewise, R.dw.e and R. dw.l stand for that of the delivery window. In practice, a rider only needs to explicitly indicate R.d and R.dw.l, as most information of a ride request can be automatically obtained from a rider's mobile phone, e.g., R.o and R.t. In addition, we can assume that both R.pw.e and R.dw.e equals to R.t, and R.pw.l can be easily obtained by adding a fixed value, e.g., 5 minutes, to R.pw.e.

B. Taxi Status

A taxi status V represents an instantaneous state of a taxi and is characterized by the following fields.

- V.ID. The unique identifier of the taxi.
- V.t. The time stamp associated with the status.
- V.l. The geographical location of the taxi at V.t.
- V.s. The current schedule of V, which is a temporally ordered sequence of origin and destination points of n ride requests R1, R2; . . .Rn such that for every ride request Ri, i =1, . . . ;n, either 1) Ri.o precedes Ri.d in the sequence (referred to as the precedence rule thereafter), or 2) only Ri.d exists in the sequence.
- V.r. The current projected route of V, which is a sequence of road network nodes calculated based on V.s.

C. Constraints

The crux of the taxi-sharing problem is to dispatch taxis to ride requests, subject to certain constraints. We say that a taxi status V satisfies a ride request R is satisfied by V if the following constraints are met.

- *Vehicle capacity constraint:* The number of riders that sit in the taxi does not exceed the number of seats of a taxi at any time.
- *Time window constraints:* All riders that are assigned to V should be able to depart from the origin point and arrive at the destination point during the corresponding pickup and delivery window, respectively.
- *Monetary constraints:* These constraints provide certain monetary incentives for both taxi drivers and riders. That is, a rider does not pay more than without taxi-sharing; a taxi driver does not earn less than without taxi-sharing when travelling the same distance; the fare of existing riders decreases when a new rider joins the trip.

III. Objective Function and Problem Definition

Since multiple taxi status may satisfy a ride request, an objective function is usually applied to find the optimal taxi. A variety of objective functions have been used in the existing literature, where a weighted cost function combining multiple factors such as travel distance increment, travel time increment and passenger waiting time, is the most common [4], [5], [6]. In this study, given a ride request, we aim to find the taxi status which satisfies the ride request with minimum increase in travel distance, formally defined as follows: given a fixed number of taxis traveling on a road network and a sequence of ride requests in ascending order of their submitted time, we aim to serve each ride request R in the stream by dispatching the taxi V which satisfies R with minimum increase in V 's scheduled travel distance on the road network.

This is obviously a greedy strategy and it does not guarantee that the total travel distance of all taxis for all ride requests is minimized. However, we still opt for this definition due to two major reasons. First, the real-time taxi-sharing problem inherently resembles a greedy problem. In practice, taxi riders usually expect that their requests can be served shortly after the submission. Given the rigid real-time context, the taxi-sharing system only has information of currently available ride requests and thus can hardly make optimized schedules based on a global scope, i.e., over a long time span. Second, the problem of minimizing the total travel distance of all taxis for the complete ride request stream is NP-complete.

IV. System Architecture

The architecture of our system is presented in Fig. 1. The cloud consists of multiple servers for different purposes and a monitor for administers to oversee the running of the system. Taxi drivers and riders use the same smart phone App to interact with the system, but are provided with different user interfaces by choosing different roles.

As shown by the red broken arrow (a), a taxi automatically reports its location to the cloud via the mobile App when (i) the taxi establishes the connection with the system, or (ii) a rider gets on and off a taxi, or (iii) at a frequency (e.g., every 20 seconds) while a taxi is connected to the system. We partition a city into disjoint cells and maintain a dynamic spatio-temporal index between taxis and cells in the indexing server, depicted as the broken arrow (b).

Denoted by the solid blue arrow 1, a rider submits a new ride request R to the Communication Server. The

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 6, June 2016

corresponding interface on a rider's smart phone where the blue pin stands for the current location of the rider. All incoming ride requests of the system are streamed into a queue and then processed according to the first-come-first-serve principle. For each ride request R, the communication server sends it to the Indexing Server to search for candidate taxis S_V that are likely to satisfy R, depicted as the blue arrow 2 . Using the maintained spatio-temporal index, the indexing server returns S_V to the communication server, denoted by the blue arrow 3.

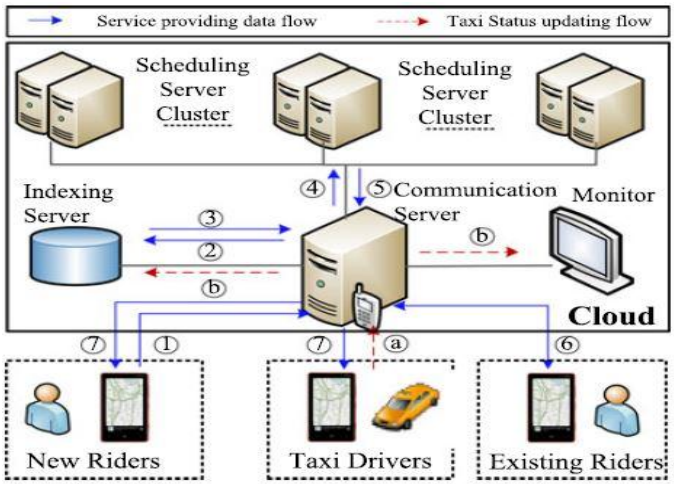

Fig.1. The architecture of the real-time taxi-sharing system.

Represented by the blue arrow 4, the communication server sends ride request R and the received candidate taxi set S_V to the Scheduling Server Cluster. The scheduling cluster checks whether each taxi in S_V can satisfy R in parallel and returns the qualified taxi V that results in minimum increase in travel distance and a detailed schedule, shown as arrow 5.

Note that in our current implementation, a single machine instead of a cluster is used to implement the indexing server and the scheduling servers. The single-machine implementation is able to answer a query in a few milliseconds, i.e., millions of queries per hour. But we believe in that the cluster- based implementation can provide a more robust service. For example, it prevents system crash in case of unexpected power failure of a single machine.

V. TAXI SEARCHING

Index of Taxis

The spatio-temporal index of taxis is built for speeding up the taxi searching process. Specifically, we partition the road network using a grid. (Other spatial indices such as R tree can be applied as well, but we envision that the high dynamics of taxis will cause prohibitive cost for maintaining such an index.). We choose the road network node which is closest to the geographical center of the grid cell as the anchor nod of the cell. The anchor node of a grid cell gi is thereafter denoted by ci. We compute the distance, denoted by dij, and travel time, denoted by tij, of the fastest path on the road network for each anchor node pair ci and cj.

Both the distance and travel time is only computed once. (Alternatively, travel time can be updated dynamically, i.e. calculated once in a while (e.g., every 10 minutes) by leveraging historical data archives, and travel time estimation techniques e.g., TDrive [7], [8], [9]. However since the effectiveness of travel time estimation is not a focus of this work, we do not discuss it in details here.) Intuitively, we can use the computed travel time to quickly filter out a large number of taxis whose schedule is "far away" from a given ride request. The distance and travel time results are saved in a matrix. The matrix is thereafter referred to as the grid distance matrix.

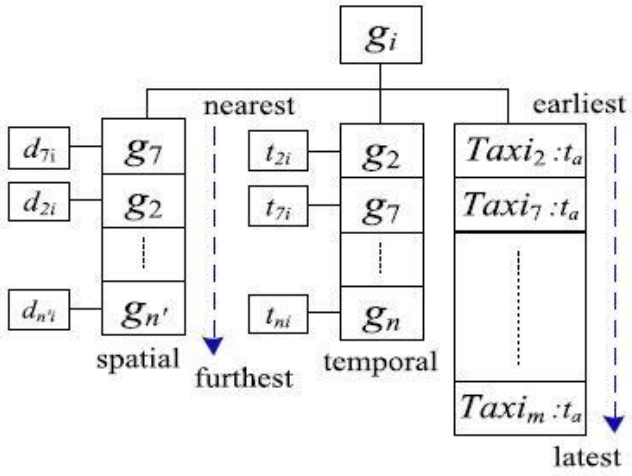

Fig.2. Spatio-temporal index of taxis

Single-Side Taxi Searching

Now we are ready to describe our first taxi searching algorithm. For the sake of the clarity of description, please consider the example shown in Fig.3. Suppose there is a request R and the current time is t_{cur}.g_7 is the grid cell in which R:o is located. g_7's temporally-ordered grid cell list $g_7.l^t_g$. g_7 is the first grid cell selected by the algorithm. Any other arbitrary grid cell gi is selected by the searching algorithm if and only if Eq. (1) holds, where t_{i7} represents

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 6, June 2016

the travel time from grid cell gi to grid cell g7. Eq. (1) indicates that any taxi currently within grid cell gi can enter g7 before the late bound of the pickup window using the travel time between the two grid cells (if we assume that each grid cell collapses to its anchor node)[7][8][9]

$$t_{cur} + t_i <= R.pw.l \quad (1)$$

To quickly find all grid cells that hold Eq. (1), the single-side searching algorithm simply tests all grid cells in the order preserved list $g_7.l^t_c$ and finds the first grid cell g_f which fails to hold Eq. (1). Then all taxis in grid cells before g_f in list $g_7.l^t_c$ are selected as candidate taxis.

In Fig.2, grid cell g_3, g_5 and g_9 are selected by the searching algorithm. Then for each selected grid cell g_s, the algorithm selects taxis (in $g_s.l_v$) whose ta is no later than $R._{wp.l}$. t_s7.

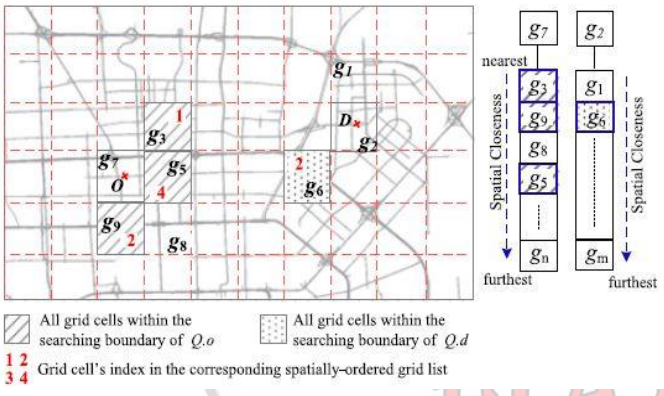

Fig. 3. Overview of the dual-side taxi searching algorithm.

Dual-Side Taxi Searching:

The dual-side searching is a bi-directional searching process which selects grid cells and taxis from the origin side and the destination side of a query simultaneously.

To dive into the details of the algorithm, consider the ride request illustrated in Fig. 2 where g7 and g2 are the grid cells in which R.o and R.d are located respectively. Squares filled with stripes stand for all possible cells searched by the algorithm at R.o side. These cells are determined by scanning g7.ltc, the temporally-order grid cell list of g7. That is, each grid cell in g7.lt c which holds Eq. (2) is a candidate cell to be searched at the origin side. Eq. (2) indicates that any taxi currently within grid cell gi can enter g7 before the late bound of the pickup window using the latest travel time between the two grid cells. The red number in each such grid cell indicates its relative position in g7.lsc , the spatially ordered grid list of g7

$$t_{cur} + t_i <= R.pw.l \quad (2)$$

Squares filled with dots indicate the candidate grid cells to be accessed by the searching algorithm at R.d side. Likewise, each such grid cell g_j is found by scanning g2.ltc to select all grid cells which holds Eq. (3), which indicates that any taxi currently in g_j can enter the g2 before the late bound of the delivery window (assuming that each grid cell collapses to its anchor node). In this example, g6 is the only satisfying grid cell as shown by Fig. 2

$$t_{cur} + t_j <= R.pw.l \quad (3)$$

Taxi Scheduling

Given the set of taxi statuses S_V retrieved for a ride request R by the taxi searching algorithm, the purpose of the taxi scheduling process is to find the taxi status in S_V which satisfies R with minimum travel distance increase.

Algorithm: Compute schedule and route after an insertion

1. Input: Ride request R, taxi status V, insertion position i for R.o, insertion position j for R.d, current time T_cur
3. If the time delay incurred by the insertion of Q.o causes the slack time of any point after position i in schedule s smaller than 0 then
4. return False
5. If the time delay incurred by the insertion of Q.o causes the slack time of any point after position i in schedule s smaller than 0 then
6. return False
7. new_schedule←insert R.o into V.s at position i
8. t_j← the scheduled arrival time of the j^{th} point of V.s
9. If $t_j+(l_j \rightarrow R.d)>R.dw.l$ then
10. return False
11. If the time delay incurred by the insertion of R.d causes the slack of any point after position j in new_schedule s smaller than 0 then
12. return False
13. new_schedule ← insert R.d into new_schedule at position j
14. return new_schedule

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 6, June 2016

Monetary Constraints

The new schedule after the insertion, so far, has only been checked against the capacity and time window constraints. It should also meet the monetary constraints. In this section we formulate the monetary constraints of taxi-sharing.

On one hand, we impose two constraints which encourage riders to participate in taxi-sharing by rewarding them with certain monetary gains. The first rider monetary constraint says that any rider who participates in taxi-sharing should pay no more than what she would pay if she takes a taxi by herself. The second rider monetary constraint says that if an occupied taxi V is to pick up a new rider Q, then each rider P currently sitting in V whose travel time is lengthened due to the pickup of Q, should get a decrease in taxi fare; and the fare decrease should be proportional to P's increase in travel time.

On the other hand, we enforce one constraint which gives the driver motivation to participate in taxi-sharing. This constraint says that a driver should charge for all distances she has travelled. Intuitively the driver should make money for the distance of reroutes incurred by the join of any new passenger.

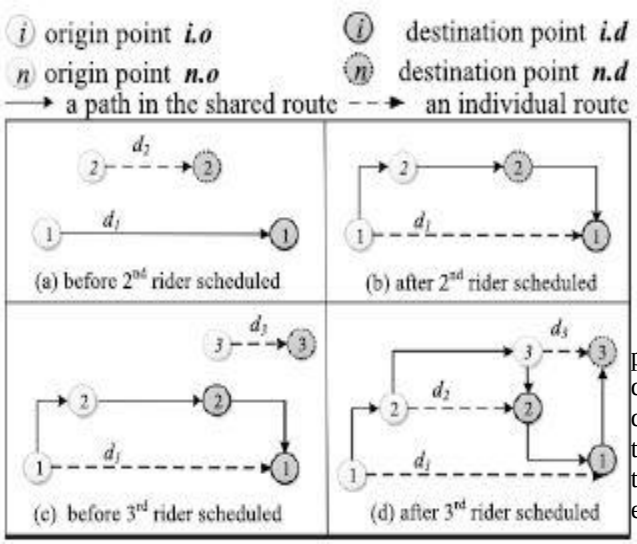

Fig. 4 An example of the pricing constraint

VI. EXPERIMENTAL RESULTS

In order to validate our proposed system under practical settings of real taxi dispatching service. We made experimental of collecting taxi service booking history during a period of 31 days spanning from December 1 to December 31 in the year of 2015. We processed that data under our proposed system. The experimental results demonstrated the effectiveness and efficiency of our system in serving real-time ride requests. Firstly, our system can enhance the delivery capability of taxis in a city so as to satisfy the commute of ore people. Secondly, the system saves the total travel distance of taxis when delivering passengers. Using the proposed monetary constraints, the system guarantees that any rider that participates in taxi-sharing saves 8% fare on average. And this system saves 11% of total distance travel of taxis on average.

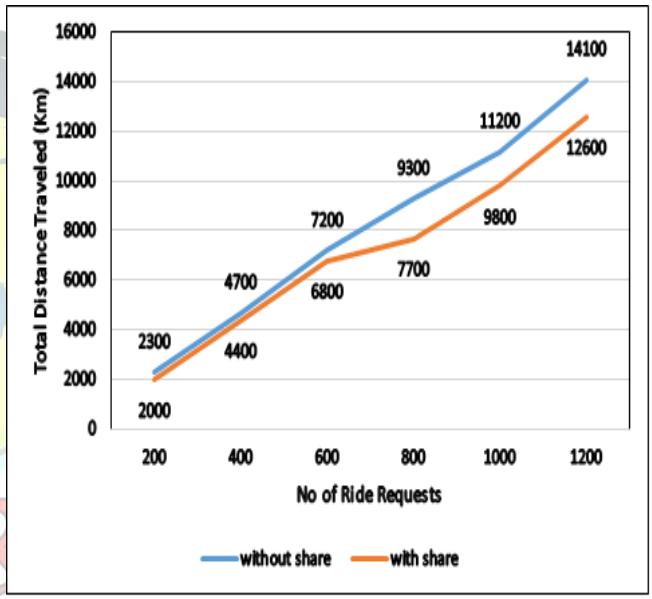

Fig. 5 Performance in effective measurements of system

VII. RELATED WORKS

Though real-time taxi-sharing has been studied in several previous works [4], [10], [11], [12], [13], our work demonstrates some major advantages. First, our problem definition is more realistic by considering three different types of constraints. Some existing works did not consider time window constraint and none of these previous works explicitly modelled monetary constraint.

VIII. CONCLUSION

This proposed system is developed a mobile cloud based real time taxi ride sharing system. It is very necessary to our country because of huge population. It will reduce the traffic on road, reduce the air pollution and fuel consumption because of taxi sharing system. If this system reduces very less percentage of these will be a considerable amount of

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Issue 6, June 2016

effect to our country. We use fuel of 4.3 billion barrels per day [2]. If this system save 1% means, even this 1% will be huge amount of savings. This system benefits in both economically and socially.

REFERENCES

[1] Indianexpress, "Delhi's odd even rule ends today: A look-back at the last 15 days", Available: http://indianexpress.com/article/cities/delhi/delhis-odd-even-rule-ends-today-a-look-back-at-the-last-15-days, [Accessed: jan. 15, 2016]

[2] Wikipedia, "Energy policy of India", Available: ttps://en.wikipedia.org/wiki/Energy_policy_of_India, [Accessed: jan. 15, 2016

[3] TimesofIndia, "India loses Rs 60,000 crore due to traffic congestion: Study", Available: http://timesofindia.indiatimes.com/india/India-loses-Rs-60000-crore-due-to-traffic-congestion-Study/articleshow/13678560 .cms , [Accessed May. 12, 2012]

[4] E. Kamar and E. Horvitz, "Collaboration and shared plans in the open world: Studies of ridesharing," in Proc. 21st Int. Jont Conf. Artif. Intell., 2009, pp. 187–194.

[5] K. Wong, I. Bell, and G. H. Michael, "Solution of the dial-a-ride problem with multi-dimensional capacity constraints," Int. Trans. Oper. Res., vol. 13, no. 3, pp. 195–208, May 2006

[6] Z. Xiang, C. Chu, and H. Chen, "A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints," Eur. J. Oper. Res., vol. 174, no. 2, pp. 1117–1139, 2006.

[7] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: Driving directions based on taxi trajectories," in Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2010, pp. 99–108.

[8] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2011, pp. 316–324

[9] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in Proc. 20th ACM SIGKDD (KDD '14), ACM, New York, NY, USA, 2014, pp. 25–34.

[10] P.-Y. Chen, J.-W. Liu, and W.-T. Chen, "A fuel-saving and pollution-reducing dynamic taxi-sharing protocol in VANETs," in Proc. IEEE 72nd Veh. Technol. Conf., Sep. 2010, pp. 1–5.

[11] P. M. d'Orey, R. Fernandes, and M. Ferreira, "Empirical evaluation of a dynamic and distributed taxi-sharing system," in Proc. 15th Int. IEEE Conf. Intell. Transp. Syst., Sep. 2012, pp. 140–146.

[12] G. Gidofalvi, T. B. Pedersen, T. Risch, and E. Zeitler, "Highly scalable trip grouping for large-scale collective transportation systems," in Proc. 11th Int. Conf. Extending Database Technol.: Adv. Database Technol., 2008, pp. 678–689.

[13] S. Ma and O. Wolfson, "Analysis and evaluation of the slugging form of ridesharing," in Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2013, pp. 64–73.