ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Online Mobile Application Using City Scale Taxi Ride Sharing

Dr.P.Selvi Rajendran, M.Vetriselvi

Professor & Head, Department of CSE ,Tagore Engineering College, India

M.E Student, Tagore Engineering College, Chennai, India

ABSTRACT: A taxi-sharing system that accepts taxi passengers' real-time ride requests sent fromsmartphones and schedules proper taxis to pick up them via ridesharing, subject to time, capacity, and monetary constraints. Themonetary constraints provide incentives for both passengers and taxi drivers: passengers will not pay more compared with no ridesharing and get compensated if their travel time is lengthened due to ridesharing; taxi drivers will make money for all the detour distance due to ridesharing. we devise a mobile-cloudarchitecture based taxi-sharing system. Taxi riders and taxi drivers use the taxi-sharing service provided by the system via a smartphone App. The Cloud first finds candidate taxis quickly for a taxi ride request using a taxi searching algorithm supported by a 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. We built an experimental platform using the GPS trajectories generated by over 33,000 taxis over a period of three months.

KEYWORDS: Spatial databases and GIS, taxi-sharing, GPS trajectory, ridesharing, urban computing, intelliegent transportation systems

I. INTRODUCTION

Taxi is an important transportation mode between public and private transportations, deliveringmillion0fpassengers 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). real-time taxi-sharing has not been wellexplored, though ridesharing based on private cars, oftenknown as carpooling or recurring ridesharing, was studiedfor years to deal with people's routine commutes, e.g., fromhome to work [1], [2]. In contrast to existing ridesharing, real-time taxi-sharing is more challenging because both ride requests and positions of taxis are highly dynamic and difficult to predict. First, passengers are often lazy to plan a taxi trip in advance, and usually submit a ride request shortly before the departure. Second, a taxi constantly travels on roads, picking up and dropping off passengers. Its destination depends on that of passengers, while passengers could go anywhere in a city.we report on a system based on the mobile cloud architecture, which enables real-time taxi-sharing in a practical setting. In the system, taxi drivers independently determine when to join and leave the service using an App installed on their smartphones. Passengers submit real-time ride requests using the same App (if they are willing to share the ride with others). Each ride request consists of the origin and destination of the trip, time windows constraining when the passengers want to be picked up and dropped off in most case, the pickup time is present.

II. RELATED WORKS

The taxi-sharing problem can be viewed as a special member of the general class of the dial-a-ride problem (DARP). The DARP is originated from and has been studied in various transport scenarios, notably goods transport [18], paratransit for handicapped and elderly personnel [19], etc. Existing works on the DARP have primarily focused on the static DARP, where all customer ride requests are known in priori. Since the general DARP is NP-hard, only small instances (involving only a few cars and dozens of ride requests) can be solved optimally (often by resorting to

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

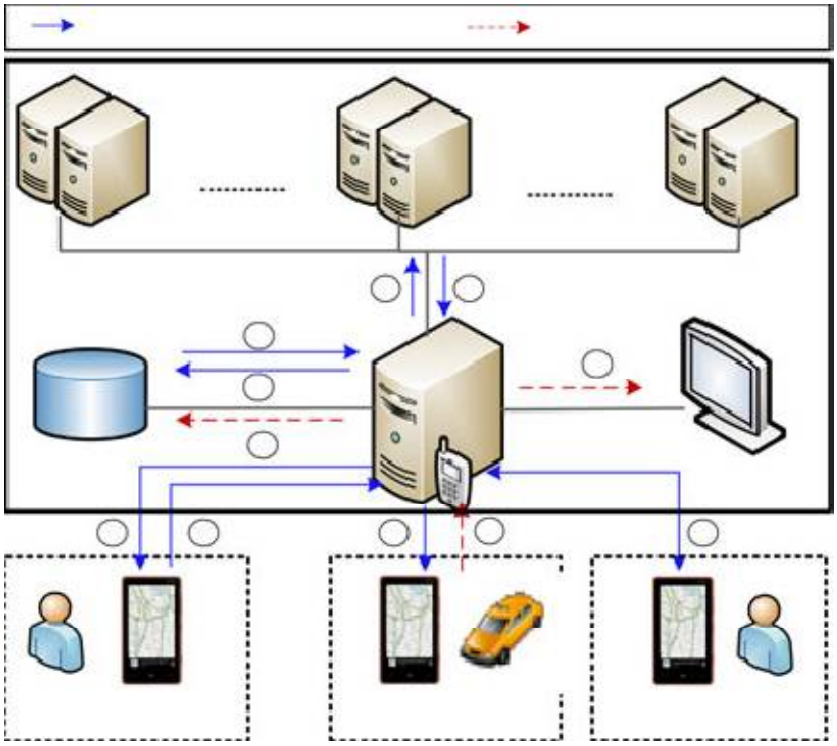
(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

integer programming techniques, see [20], [21]). Large static DARP instances are usually solved by using the two-phase scheduling strategy [6], [22], [23], [24] with heuristics. Specifically, the phase I partitions ride requests into some group and computes an initial schedule for delivering the riders in each group. In phase II, ride requests are swapped between different groups, aiming to find new schedules optimizing a predefined objective function. Little research has been carried out on the dynamic DARP, where requests are generated on the fly. Previous works on the dynamic DARP problem [25] continues to adapt the two-phase scheduling strategy. However, the two-phase strategy is not feasible for the real-time taxisharing

III. FIGURE SYSTEM ARCHITECTURE

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 (detailed in Section 4.1), depicted as



the broken arrow (b). Denoted by the solid blue arrow_1 , a rider submits a new ride request Q to the Communication Server. Fig. 2b shows the 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 Q, the communication server sends it to the Indexing Server to search for candidate taxis SV that are likely to satisfy Q, depicted as the blue arrow _2 . Using the maintained spatio-temporal index, the indexing server returns SV to the communication server, denoted by the blue arrow_3 .

IV. THE PROPOSED SCHEME: SEARCHING ALGORITHMS AND SCHEDULING ALGORITHMS

The taxi searching module quickly selects a small set of candidate taxis with the help of the spatio-temporal index. In this section, we will first describe the index structure and then detail the searching algorithm.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

a. 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 . Suppose there is a query Q and the current time is tcur : g7 is the grid cell in whichQ:o is located. g7‘s temporally-ordered grid cell list g7 :lt g is shown on the right of . g7is 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 ti7 represents the travel time from grid cell gi to grid cell g7. Eq. (1) indicates that any taxi currently within grid cell gican 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) ti7 þ tcur _ Q:pw: l (1) To quickly find all grid cells that hold Eq. (1), the single-side searching algorithm simply tests all grid cells in the orderpreserved list g7 : lt c and finds the first grid cell gf which fails to hold Eq. (1). Then all taxis in grid cells before gf in list g7 :lt c are selected as candidate taxis. In , grid cell g3, g5 and g9 are selected by the searching algorithm. Then for each selected grid cell gs, the algorithm selects taxis (in gs : lv) whose ta is no later than Q: wp:l _ ts7. For instance, Fig. 8 shows how taxis are selected from grid cell g7 and g3. The taxi which can satisfy Q with the smallest increase in travel distance must be included in one of the selected grid cells (under the assumption that each grid cell collapses). Unfortunately, this algorithm only considers taxis currently “near” the pickup point of a query (thus called single-side search). As the number of selected grid cells could be large, this algorithm may result in many taxis retrieved for the later scheduling module (therefore increasing the entire computation cost), which is certainly not desirable for a rigid realtime application like taxi ridesharing.

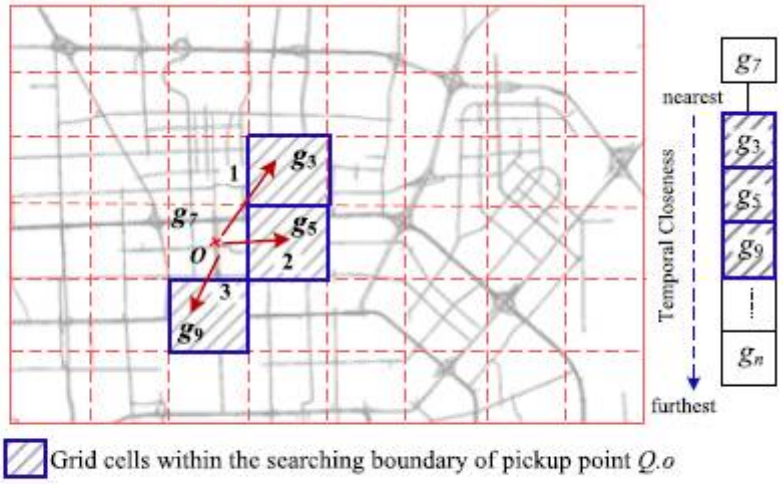


Fig. 7. The single-side taxi searching algorithm.

b. 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.These cells are determined by scanning g7 :lt c, 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 (assuming each grid cell collapses to its anchor node). The red number in each such grid cell indicates its relative position in g7 :lsc, the spatiallyordered grid list of g7

$$t_{cur} + t_{i7} \leq Q.dw.l. \qquad (2)$$

Squares filled with dots indicate the candidate grid cells to be accessed by the searching algorithm at Q:d side. Likewise,each such grid cell gj is found by scanning g2 : lt c to select all grid cells which holds Eq. (3), which indicates that any taxi currently in gj can enter the g2 before the late boundof the delivery window (assuming that each grid cell collapse to its anchor node). In this example, g6 is the only satisfyinggrid cell as shown by

$$t_{cur} + t_{j2} \leq Q.dw.l. \qquad (3)$$

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

then illustrates the searching process step by step. The algorithm maintains a set So and a set Sd to store the taxis selected from Q:o side and Q:d side respectively. Initially, both So and Sd are empty. The first step in the searching is to add the taxis selected from taxi list g7 : lv to taxi set So as depicted in , and add the taxis selected from taxi list g2 : lv to taxi set Sd as depicted by 10b. Then thealgorithm calculates the intersection of So and Sd.

c. TimeWindowConstraints:

A schedule of n points, there is clearly O n2 ð Þ ways to insert a new ride request into the schedule. To insert Q3:o after point Q1:o optimally, the algorithm needs to find the first path (starting from the shortest path) from Q1 : o to Q3 : o which allows the taxi to arrive at Q3 : o during Q3 : pw given the scheduled arrival time at Q1 : o. Since the shortest path is often not the fastest one when considering real road traffic, it is likely that multiple paths needs to be calculated before finding the first satisfactory path from Q1 : o to Q3 : o: Denote by ! the travel time of the fastest path from one location to another location, and twrepresents the time spentwaiting for the passenger if the taxi arrives Q3 : o ahead of Q3 : pw : e. Eq. (4) gives the travel time delay, denoted by td after insertingQ3 : o betweenQ1 : o and Q2 : o

$$t_d = (Q_1 . o \rightarrow Q_3 . o) + (Q_3 . o \rightarrow Q_2 . o) + t_w - (Q_1 . o \rightarrow Q_2 . o) \quad (4)$$

d. Monetary Constraints

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. a taxi status V anda new ride request Qn, under what conditions will V satisfy the above three monetary constraints with respect to Qn.Denote by Q1; . . .Qn_1 the riders involved in the current schedule of V before the join of Qn. Also denote by di the distance between Qi:o and Qi:d on the road network; i ¼ 1; . . . ; n. Denote by fi the taxi fare of rider Qi if V picks up Qn. Denote by F : Rþ ! Rþ the fare calculation function,which maps the travelled distance to the taxi fare. FunctionF can be defined by some transportation authority or taxi company. Then the first monetary constraint canbeexpressed by Eq. (7)

$$f_i \leq F(d_i), i = 1, \ldots, n. \quad (7)$$

Denote by M the revenue of the driver if she picks up Qn and by D the travel distance of the new route after the pickup. Then the driver monetary constraint is expressed byEq. (8)

$$M \geq F(D). \quad (8)$$

Since M ¼ Sfi, we then have Eq. (9) by bridging the twoequations above

$$F(D) \leq M = \Sigma f_i \leq \Sigma F(d_i), i = 1, \ldots, n. \quad (9)$$

We thus introduce a parameter Qi:r for each rider Qi, which presents Qi's acceptable money-to-time rate. That is to say, Qi supports the pickup of a new rider only when the ratio of the fare decrease to the travel time increase is larger than Qi:r. The above constraint is expressed by Eq. (10). And an insertion satisfies the monetary constraints only when all current riders on the taxi support the pickup decision

$$f_n = F(d_n) - f, \quad (10)$$

V. EVALUATION

a. Data Set

i) **Roadnetworks**: We perform the experiments using the realroad network of Beijing, which contain 106,579 road nodes and 141,380 road segments.

ii) **Taxi Trajectories**: The taxi trajectory data set contains the GPS trajectory of over 33,000 taxis during a period of 87 days spanning from March to May in the year of 2011.The total distance of the data set is

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

more than 400 million kilometres and the number of points reaches 790 million.After trip segmentation, there are in total 20 million trips,among which 46 percent are occupied trips and 54 percentare non-occupied trips. We map each occupied trip to the road network of Beijing using the map-matching algorithm proposed

iii) **Experimental Platform**: To validate our proposed system under practical settings, instead of generating random ride requests and initial taxi statuses, we mine the trajectory data set to build anexperimental platform. From the historical trajectory data set, the platform learns information regarding 1) the distribution of the ride requests on the road network over time of day, and 2) the mobility patterns of the taxis. With this learned knowledge, the platform then generates a realistic ride request stream (meaning that the origin-destination pairs and time windows of ride requests follow the learned distribution) and initial taxi statuses for our experiments.

iv) **Ride request stream:** The goal is to generate real-time ride requests that are as realistic as possible. For this purpose,we first discretise one day into small time frames, denoted by fj's. Denote all road segments by ri's. We assign all historical ride requests into time frames. Assume that the arrivals of ride requests on each road segment approximately follow a Poisson distribution during time frame fj. Thus,we can learn _ji , i.e., the parameter of the Poisson distribution for road segment ri during time frame fj. Specifically, for each road segment ri, we count the number of ride requests that originate from ri within time frame fj, denoted by cji . Then we calculate _j i based on cji using Eq. (14) (where lenðfÞ is the length of a frame in time units) and generate a ride request stream that follows a Poisson process with parameter _j i . In order to generate destination of requests trufhfully, for each cji , we decompose it into an array of numbers fcji1; cji2; . . . cjimg, where cjik; k ¼ 1; 2; ::m represents the number of requests which are originated from road segment ri and destined for road segment rkduring time frame fj, as illustrated by Fig. 14a. Therefore, the transition probability from ri to rk during time frame fj, denoted by pjik, can be estimated using Eq. (15). Fig. 14b shows the distribution of destination road segments for requests that originatefrom road segment riin an hour

$$\lambda_i^j = c_i^j / len(f), \qquad (14)$$

b. Baseline Methods

The Non-Taxi-sharing method (NR) forbids taxi-sharing and assumes that a vacant taxi moves to pick up the rider that it can pick up at the earliest time.

c. **Taxi searching step**: A taxi-sharing method is single-side if the taxi searching algorithm retrieves taxis only from the origin side of a request; otherwise, it is dual-side.

d. **Taxischedulingstep**: A taxi-sharing method is called bestfitwhere the taxi scheduling process tries all candidate taxis returned by the taxi searching algorithm. Otherwise, is called first-fit if the scheduling process terminates once it finds a taxi that satisfies the ride request.(SB), Dual-side and First Fit Taxi-sharing (DF), Dual-side andBest-fit Taxi-sharing (DB). The fare calculation function F ¼ pD, where D is the traveled distance and p is some constant price for a unit traveled distance. The money-to-time rates of ride requests are assumed to follow an exponential distribution with a mean value m.

VI. SIMULATION RESULTS, PERFORMANCE EVALUATION AND COMPARISON

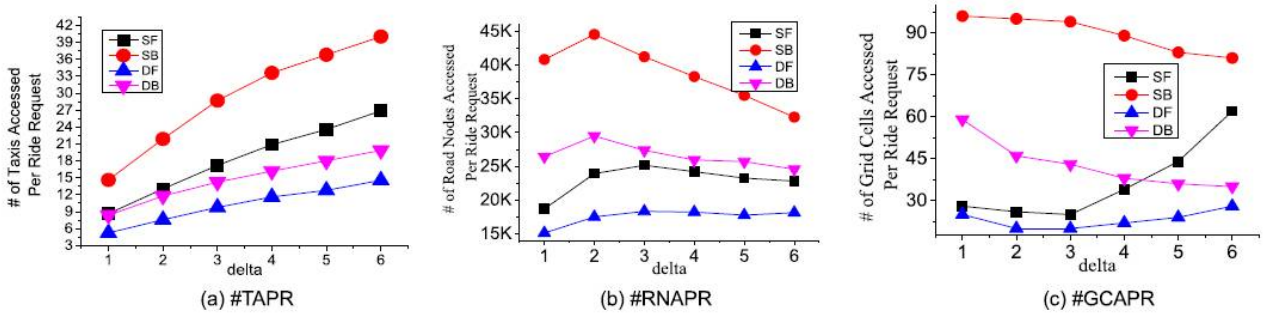
We also test the efficiency of the system using machineindependent measurements. The three sub-graphs of Fig. 18 show the number of taxis accessed per ride request, the number of road nodes accessed per ride request, and the number of grid cells accessed per ride request, respectively, for different taxi-sharing methods under different D. It is clear from the pictures that all taxi-sharing methods do not show sharp increase in computation cost as D increases. It is also obvious that the computation cost of the DB taxi-sharing method is significantly smaller than that of SB taxi-sharing method. Especially when D 4, the computation cost of the DB method is even smaller than that of the SF method. The result of Figs. 16 and 18 together validate our motivation for the dual-side taxi searching algorithm. That is, the dual-side searching indeed incurs small increase in travel distance in exchange for the significant decrease in computation cost.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

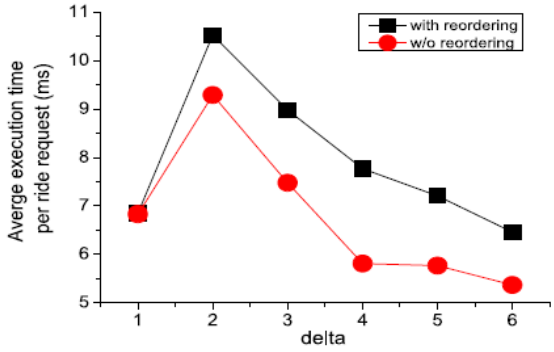
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016



Computational cost in terms of node access per ride request



Time cost of schedule reordering

VII. CONCLUSION

Firstly, our system can enhance the delivery capability of taxis in a city so as to satisfy the commute of more people. For instance, when the ratio between the number of taxi ride requests and the number of taxis is 6, our proposed system served three times as many ride requests as that with no taxi-sharing. Secondly, the system saves the total travel distance of taxis when delivering passengers, e.g., itsaved 11 percent travel distance with the same ratio mentioned above. Supposing a taxi consumes 8 liters of gasoline per 100 km and given the fact learned from the real trajectory data set that the average travel distance of a taxi in a day in Beijing is about 480 km, the system can save over one third million liter of gasoline per day, which is over 120 million liter of gasoline per year (worth about 150million dollar).
Thirdly, the system can also save the taxi fare for each individual rider while the profit of taxi drivers does not decrease compared with the case where no taxi-sharing is conducted. Using the proposed monetary constraints, the system guarantees that any rider that participates in taxi-sharing saves 7 percent fare on average. In addition, the experimental results justified the importanceof the dual-side searching algorithm. Compared to the single-side taxi searching algorithm, the dual-side taxi searching algorithm reduced the computation cost by over 50 percent, while the travel distance was only about 1 percent higher on average.
The experimental results also suggest that reordering the points of a schedule before the insertion of the new ride request is not necessary in practice for the purpose of travel distance minimization.In the future, we consider incorporating the creditability of taxi drivers and riders into the taxi searching and scheduling algorithms. Additionally, we will further reduce the travel distance of taxis via ridesharing. We also consider refining the ridesharing model by introducing social constraints, such as gender preference, habits preference (e.g., some people may prefer co-passengerswho do not smoke).

REFERENCES

[1] R. Baldacci, V. Maniezzo, and A. Mingozzi, “An exact method for the car pooling problem based on lagrangean column generation,” Oper.Res., vol. 52, no. 3, pp. 422–439, 2004.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

[2] R. W. Calvo, F. de Luigi, P. Haastrup, and V. Maniezzo, "A distributed geographic information system for the daily carpooling problem," Comput.Oper.Res., vol. 31, pp. 2263–2278, 2004.
[3] S. Ma, Y. Zheng, and O. Wolfson, "T-Share: A large-scale dynamic ridesharing service," in Proc. 29th IEEE Int. Conf. Data Eng., 2013, pp. 410–421.
[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 knowledgefrom the physical world," in Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2011, pp. 316–324.
[9] O. Wolfson, A. P. Sistla, B. Xu, J. Zhou, S. Chamberlain, Y. Yesha, and N. Rishe, "Tracking moving objects using database technology in DOMINO," in Proc. 4th Int. Workshop Next Generation Inf. Technol. Syst., 1999, pp. 112–119.
[10] J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G.-Z. Sun, "An interactive-voting based map matching algorithm," in Proc. 11th Int. Conf. Mobile Data Manage., 2010, pp. 43–52.
[11] J. Yuan, Y. Zheng, L. Zhang, Xi. Xie, and G. Sun, "Where to find my next passenger," in Proc. 13th Int. Conf. Ubiquitous Comput., 2011, pp. 109–118.
[12] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, "An energy-efficient mobile recommender system," in Proc. 16thACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 899–908.
[13] R. Balan, K. Nguyen, and L. Jiang, "Real-time trip information service for a large taxi fleet," in Proc. 9th Int. Conf. Mob. Syst. Appl. Serv., 2011, pp. 99–112.
[14] K. Yamamoto, K. Uesugi, and T. Watanabe, "Adaptive routing of cruising taxis by mutual exchange of pathways," in Proc. 12th Int. Conf. Knowledge-Based Intell. Inf. Eng. Syst., Part II, 2008, pp. 559–566.
[15] D. Santani, R. K. Balan, and C. J. Woodard, "Spatio-temporal efficiency in a taxi dispatch system," Research Collection School Of Information Systems, Singapore Management University, Oct. 2008.
[16] D. Zhang and T. He, "CallCab: A unified recommendation system for carpooling and regular taxicab services," in Proc. IEEE Int. Conf. Big Data, 2013, pp. 439–447.
[17] W. Wu,W. S. Ng, S. Krishnaswamy, and A. Sinha, "To Taxi or Not to Taxi?—Enabling personalised and real-time transportation decisions for mobile users," in Proc. IEEE 13th Int. Conf. Mob. Data Manage., Jul. 2012, pp. 320–323.
[18] Y. Dumas, J. Desrosiers, and F. Soumis, "The pickup and delivery problem with time windows," Eur. J. Oper. Res., vol. 54, no. 1, pp. 7–22, Sep. 1991.
[19] A. Beaudry, G. Laporte, T. Melo, and S. Nickel, "Dynamic transportationof patients in hospitals," OR Spectr., vol. 32, no. 1, pp. 77–107,2010.

BIOGRAPHY

Dr.P.Selvi Rajendran Professor& Head in the Computer Science Department, Tagore Engineering College Anna University. Her research interests are Data Mining,online Mobile Application using searching and scheduleing Algorithms.

M.Vetri Selvi Student of Tagore Engineering College, Anna University pursing Post Graduation Master of Engineering in Computer Science. Her research in online mobile application using Taxi ride sharing