ISSN:0975-9646

Real Time Taxi Ride Sharing

Prof. Deeksha Bhardwaj, Azam Khan, Sunny Patil, Rajat Dhoot

GHRIET, Computer Engineering Department,
Savitribai Phule Pune University Pune 412207, India

***Abstract*— The traffic congestion has been increasing in urban areas due to this studying alternative measure of mobility management, and one of these measures is carpooling. In theory, these systems could lead to a reduction in the use of private vehicles to achieve success there are limitations because of two reasons the psychological barriers associated with riding with strangers and the flexibility of poor scheduling. The limitation of traditional scheme has been overcome by studying a model of carpooling with this model new feature is introduced Establishing a base trust level for Carpoolers to find suitable matches for traditional group and at the same time allowing to search for a ride with the other alternatives group when numbers of pool has scheduled trip different from usual one. People are migrating from one place to another but due to insufficient public transport facilities people preferred to own vehicles. This lead to problem like number of vehicles traffic, fuel consumption, air pollution, parking problem and increase in overall expenses by using unique vehicles per user. To overcome this hurdle real time carpooling can be used. Concept of car polling is to share same vehicles by the passenger travelling in the same route instead of travelling by personal vehicles. This help us to resolve the problem of traffic jam fuel consumption and also controlling air pollution result in green environment. The android application will successful overcome the problem as android mobiles are easily accessible, available and user friendly to everyone**

I. Introduction

Carpooling is an activity that enables its pursuers to abandon their individual means of transport, to SHARE a single vehicle for traveling. This affects the pollution and fuel consumption levels by ensuring a reduction in them and in the process relieves of the road from vehicular congestion. This concept is popular in countries like U.S.A and Canada to a considerable extent. The State takes care to encourage the public to carpool. In China, during the 2008 Beijing Olympics, there were instances of the citizens resorting to it when there were a lot of restrictions imposed on the vehicles plying on the roads every day. I was curious to find out the extent of its popularity in India.

A simple Google search revealed an array of websites that offer wide variety of services to facilitate carpooling. I could not make a conclusive judgment on the success rate of these endeavors with respect to the number of car users in India. Since I could not find any other reliable information regarding the number of people carpooling in the country on any given day, I was left with just the option of accepting the claims made by these sites. It would at least save me from the tedium of verifying its authenticity. But I can tell you it is still very less compared to the car users in India.

If we were to list out the sustainable ways of transportation, carpooling can be definitely considered to be a viable and comfortable option for those who are not accustomed to the rigors of a public transport system. Calling it a money-saving option would be like thinking along the lines of Havells ad. But, in reality, it is indeed a fuel and expenditure saving option and that the activity requires minimum efforts or infrastructure should be an added bonus. Although it could take some time for any "laws" related to carpooling to come into effect, it is up to the citizens to take it seriously whenever and wherever possible so that it culminates into reduction in pollution and fuel consumption and ultimately saves the environment from destruction.

Taxi is an important transportation mode between commercial and private transportation, delivering millions of passengers to different locations in urban areas. However, the number of taxi is much less than its demand in peak hours of major cities, due to this many people stand at roadside waiting for the taxis. To overcome the problem one optimal solution is to increase the taxis. 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. To address this issue, we introduce a carpooling system that accepts passengers' real-time ride requests sent from smartphones and schedules proper taxis to pick up them via taxi sharing with time, capacity, and monetary constraints.

II. Problem Statement

Taxi is an important transportation mode between commercial and private transportation, delivering millions of passengers to different locations in urban areas. However, the number of taxi is much less than its demand in peak hours of major cities, due to this many people stand at roadside waiting for the taxis. To overcome the problem one optimal solution is to increase the taxis. 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. To address this issue, we introduce a carpooling system that accepts passengers' real-time ride requests sent from smartphones and schedules proper taxis to pick up them via taxi sharing with time, capacity, and monetary constraints.

III. System Architecture

The System architecture show how the system actually works interacts. The main modules are the passenger and driver. The application will be installed on both the devices and they will interact with each other.The most important thing consider in carpooling application mention below:

- The application will be installed on the driver and passenger android phone.

- The id to the passenger and driver was allotted by database system
- Activities are managed and controlled by central database
- The ratings and comments history will be displayed on mobile phones.

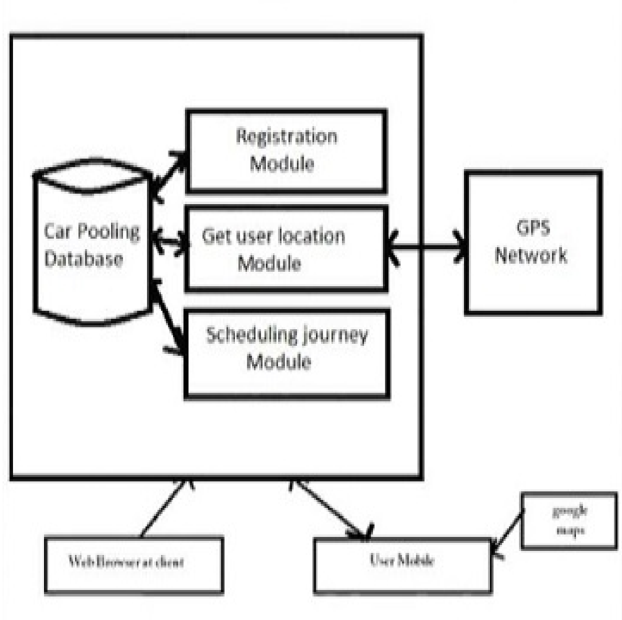

Fig.1 Carpooling architecture

A. System Design

We are going to show you how the user and passenger will communicate with each other with the help of use case diagram and class diagram. The registration is required in android application for the first time users either passenger or driver in android application .Registered users just have to login with their login credentials like username and password. The user will select weather he is driver or passenger. If he is a driver then he will update the route details (through path) by sending details to the cloud server. The details updated by the driver will be shown on the passenger's android phone. The passenger will select and book the convenient seat for travelling. Once the ride is successfully completed the driver and passenger will rate each other. Google map will guide/direct the driver for reaching the desired destination.

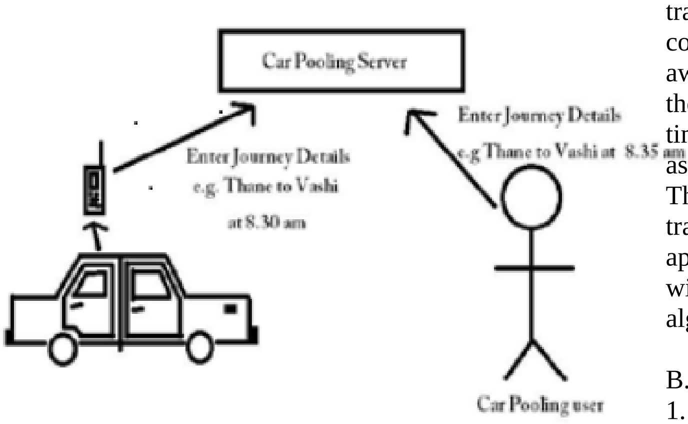

Fig .2 user requesting for pooling

1) User (driver or passenger) has to register in android application which is connected through internet to database.
2) The registered users which are logged in application can perform various activities like:-
The user can get location of car which is nearest to him.
The user will schedule his drive.
The user can select the car which is nearest to him.
The google map will provide exact car location to the user.
User can send the request to the driver for ride.
The user can also check the details of driver which is carpooling for the purpose of his security.
User can check the fare amount for his destination.
The android mobile application will do the task of carpooling server and google map.
3) After the successful completion of ride user and driver both will rate each other for future rides.

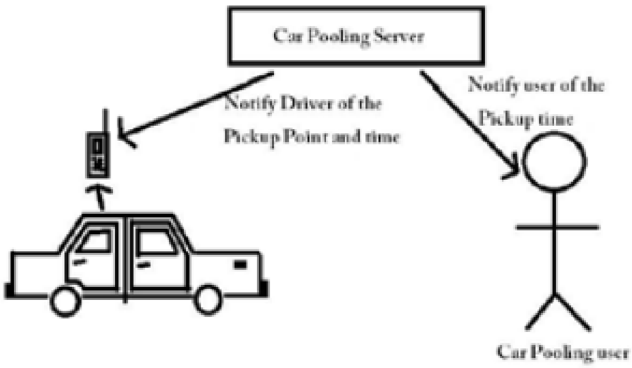

Fig.3 Driver accepting the users request

IV. TAXI SEARCHING

The spatio temporal index helps in finding the nearest set of candidate taxis.
We will first show the index structure and then the details about the searching algorithm.

A. Index of Taxis

The concept of spatio temporal index is used for taxi searching process at a high speed. With the help of grid the road networks are distributed. The road node are selected which is near to the centre of the grid cell as anchor node. The anchor node is used for calculating the distance and travel time. The travelling time is not taken into consideration for the effectiveness. The taxi which are far away from the requester are removed for that request with the use of calculated time travel. The result of calculated time travel are saved in matrix form so the matrix is known as grid distance matrix.
The two anchor node is used to find the distance and time travel of the two network node therefore this provide the appropriate distance between any two network node, this will reduce the cost of calculating path in taxi searching algorithm

B. Searching Algorithms

1. Single-Side Taxi Searching

The first searching algorithm is single side taxi searching that we will discuss. Here we will consider query as Q and the nearest grid cell will take as g. Now Q.o in g(7) is origin in side i.e from where the drive is started. Q.d in g(2). The algorithm will select the grid cell g(i) which has the travel time t(i7) from g(i). This indicate that any taxi which is present in the grid cell g(i) can reach the destination side grid cell g(7) without any delay.

$$t(i7)+t(cur)<=Q.pw.l \qquad (1)$$

To find all grid cell which are satisfying the equation (1) the algorithm will check all the grid cell in the first come first search order and when the algorithm gets the first grid cell which is not satisfying the equation (1) the algorithm will declare the taxi which is present in the grid cell before the grid cell which is not satisfying the equation (1) as candidate taxi.

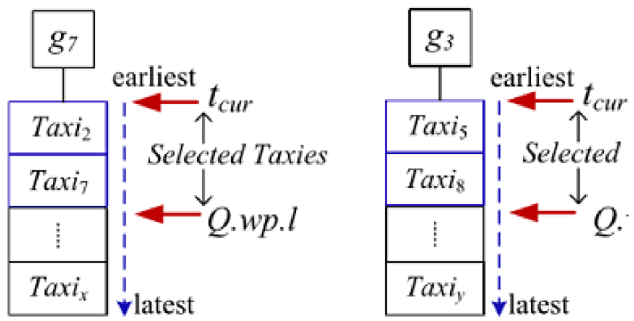

Fig.4 Choose taxis from the selected grid cells.

The taxi will be included in the grid cell which can satisfy the query Q with the little increase in the travelling distance. This algorithm is very costly as it consider the taxi which is closest to the pickup point of a query. This algorithm is not efficient for the real time carpooling application because the number of grid cell which is selected is larger this lead to the taxis retrieved scheduling module. We have the spatiotemporal factor on destination point which helps in reducing the number of grid cell which need to be get selected.

For decreasing the algorithm cost and obtain best distance we have introduce the searching algorithm from both the sides.

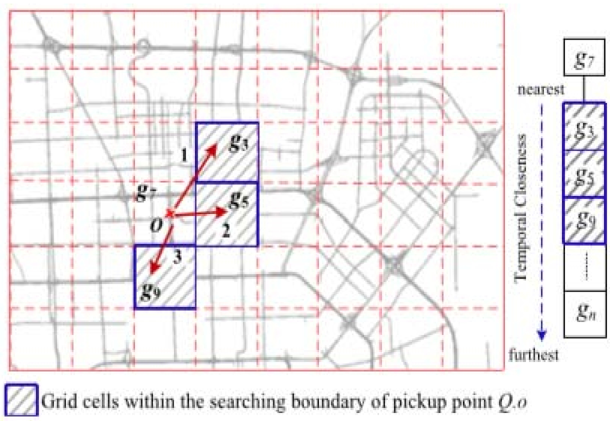

Fig . 5 The single-side taxi searching algorithm.

2. Dual-Side Taxi searching

In dual-side searching algorithm the selection of taxis is from both side i.e from the origin side and the destination side of a query and thus it is known as bi-directional searching algorithm.

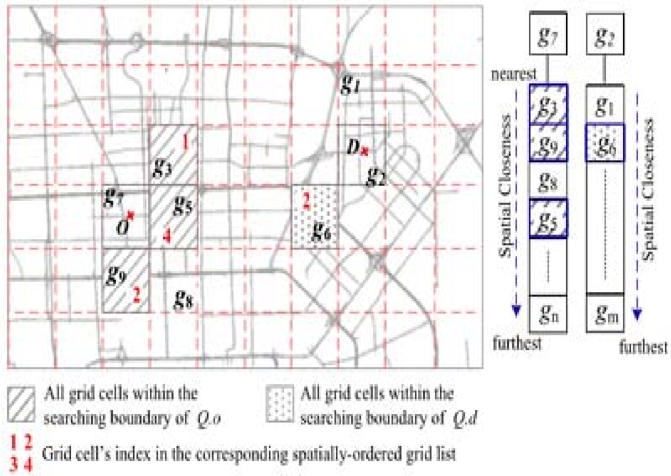

Fig. 6 The dual-side taxi searching algorithm.

Now we will take the grid cells g(7) which contain Q.o and g(2) which contain Q.d. The cells searched by algorithm at Q.o side are shown as squares filled with stripes in figure. The temporarly ordered g(7) grid cell is scanned for searching this grid cells. The grid cell which holds equation (2) indicate that this is the candidate cell from the origin side. The taxi which is present in the g(i) grid cell can come into g(7) without the delay of time.

$$t(cur)+t(i7)<=Q.dw.l \qquad (2)$$

The candidate cell which can be searched by the searching algorithm by the destination side g(2) is shown in figure as squares filled with dots. The scanning of g(2) is done for searching g(i) which shows that taxi which is present in g(i) can come in g(2) without delaying the travelling time.

$$t(cur)+t(j2)<=Q.dw.l \qquad (3)$$

3. Dijkstra's algorithm

Dutch computer scientist Edsger Dijkstra in 1956 invent Dijkstra’s algorithm and published in 1959, the single source shortest path problem is solved by this graph based algorithm. It is applied only on positive weights graphs. In routing this algorithm is often used. Dijkstra’s algorithm is used for finding the shortest path with minimum cost. “For example:-Let vertices in a graph the location point &edges which link these vertices are the driving distances from one location to another location. DSPA is used to find the shortest route from one city to another with minimum cost. It solves only the problems with non-negative costs, i.e., Cij >=0 for all (i, j) belongs to E, Where C is the cost & E is the edges for a graph

V. TAXI SCHEDULING

In this algorithm we have to find the minimum travel distance increase by taking the query for given set of taxi status which satisfies the query of the user.

Basically in this algorithm we have to insert the query into the schedule of taxi status which gives the minimum increase in distance for travelling. The insertion can be done in three ways: (1) according to the precedent rule the points should be reorder in current schedule. (2) Insert Q.o in schedule (3) insert Q.d in schedule. For all insertion the insertion which has the minimum distance travel is chosen for taxi status. The time delay mechanism is check in all three insertion, if the mechanism is not satisfied it immediately stops insertion.

Consider a schedule with n points, among which m points are pickup points. After the first step, i.e., the schedule reordering step, there will be as many as n!=2m sequences which comply with the precedence rule. (n! is the total number of sequences of all n points. Since each pickup point must precede the corresponding destination point, so the total number of sequences needs to be divided by 2m.) Though reordering the schedule is theoretically necessary for finding the optimal insertion way, we find that it is not the case in practice via experiments. Therefore, for the sake of simplicity, we do not consider the schedule reordering step here. Next we describe how to check the feasibility of each insertion possibility, subject to the capacity and time window constraints first (Section 5.1) and then the monetary constraints (Section 5.2), given a pair of Q and V. A computer cluster can be employed to parallelize the computation by assigning taxi statuses to different computers in the cluster, so the constraints checking for multiple taxi statuses can be performed simultaneously.

VI. REQUIREMENTS

A. Server Hardware

1) Computer with minimum configuration of processor 1.33 GHz, 512mb RAM, 80 GB hard disk.

2) Internet Connection for server.

B. Server Software

1) Eclipse IDE(3.4) tool

2) Microsoft SQL Server

3) Xampp server which is having MySQL and php as cross platform.

C. Client Hardware

1) Android enabled mobile phone with GPS.

D. Client Software

1) Android sdk 1.2 or above.

2) Google Maps

VII. CONCLUSION

We are developing a mobile based carpooling application. This application would help the process of creation of instant carpool events. This system uses the resources and sent messages to the users as per their location. The system will save money reduce traffic congestion, help to save environment as well as reduce stress. This system is beneficial all three user (rider), administrator and retailer(businessman).

Thus, we successfully reduced the long conversations needed for normal car pool events. In future, more functionality can be added to make this application more robust and more feature rich. With the advent of smart phones, this application, when developed to its fullest, would be able for all to use and make their journeys much more enjoyable and comfortable.

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

[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 knowledge from 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.