



IMPLEMENTATION OF DYNAMIC CARPOOLING SYSTEM

Prof. Anil Hingmire | Prajakta Pednekar | Sukanya Naik | Vinay Gupta

ABSTRACT

Carpooling (also known as car-sharing, ride-sharing and lift sharing), is the sharing of car journeys so that more than one person travels in a car. By having more one vehicle, carpooling reduces each person's travels costs such as fuel costs, tolls, and the stress of driving. Carpooling is also seen as a more environmentally friendly and sustainable way to travel as sharing journeys reduces carbon emissions, traffic congestion on the roads, and the need for parking spaces. Authorities often encourage carpooling, especially during high pollution periods and high fuel prices. We intent on making an ANDROID based application that will enable to let people know if vehicles are available for carpool in their desired path they can sign in for it. This will enable people using this application to share expense, not worry about hiring a cab and making new connections. People having this application on their cell phone can easily carpool with unacquainted people without worrying about security. The front end will be developed using JAVA Android and the backend will work on MySQL database.

KEYWORDS: Car Owner, Ride Seeker, Pickup and Drop-Off points, SD [Source & Destination], GPS, Dijkstra's algorithm.

I. INTRODUCTION

Transportation is a major issue in our world today. Carpooling is a solution to the problems of traffic jams, pollution, and extra use of fuel. Our application is an attempt to make a system which is user friendly and provides an opportunity to share cars. The service will allow users to offer and request ride sharing journeys using their Android enabled phones.

The main problem in carpooling is how to find out who travels to the same destination as yours every day or who is interested in carpooling. Carpooling allows a large number of passengers and drivers to be matched with each other automatically and instantly wishing to travel same destination. Using this developed android application car owner can create a ride by giving information like source, destination, starting time of journey, available seats etc. And ride seeker will be able to search and browse the rides by providing inputs like source, destination, and time. The dynamic Carpooling system relies on the information from two users i.e. ride creator or car owner and ride seeker.

The users that have this application installed on their cell phones can use this application as either initiators of car pooling event or as invitees for a car pooling event. When the user launches this application, the user is given three options to choose from. The user can create a new car pool event, view the events that he has created earlier or view invitations for car pool event from other users. When the user chooses to create a new event, he is given a form which has fields such as subject of the car pool, time of the car pool, vacancies in the car, destination address of the car pool, and a button to choose from the list of his contacts.

II. DYNAMIC ROUTE MATCHING AMONG CAR OWNER AND RIDE SEEKER

A. Global Positioning System (GPS)

GPS is a satellite based system that can be used in navigation to locate the positions anywhere on the earth. GPS is designed & operated by U.S. Department of Defense (DOD). GPS consists of satellites, control & monitor stations and GPS receivers. GPS receivers take information which is transmitted from the satellites and uses triangulation to calculate a user's

exact location. GPS is used in a variety of ways:

- To determine the position of locations.
- To navigate from one location to another.
- To create digitized maps.
- To determine the distance between two points

In this paper we use the GPS in Dijkstra's algorithm for finding the current location. By using this position we calculate the distance from source to every node in the graph. From this we also estimate the shortest path. Distance is given by a formula:

$$\text{Distance} = [(x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2]^{1/2}$$

Where x, y, z are the coordinates of a position given by GPS.

B. Proposed Algorithm

Step1:

INITIALIZE $d[s]=0$ for all $v \in V$

$\{s\}$, where s as source, V as set of all vertices.

Do $d[v]=\infty$. //set all node's distances to ∞ except s.

Step2:

Get the current position (x1, y1) of source node from GPS.

Source_x=x1;

Source_y=y1;

Dist.=0;

Step3:

S is the set of visited vertices.

Set $S = \emptyset$ //S is initially empty.

$Q = V$ //Queue initially contain all the vertices.

While $Q \neq \emptyset$ //while Q is not initially empty.

Do $u = \text{mindistance}(Q, d)$ //select element of Q with min. distance.

$S = S \cup \{u\}$ //add u to the list of visited vertices.

Step4:

Get the position (x2, y2) of the visited nodes from GPS $\text{Current_x} = x2; \text{Current_y} = y2;$

$\text{distance} = \sqrt{(x2-\text{source_x})^2 + (y2-\text{source_y})^2}$

$\text{dist} = \text{distance} + \text{dist}$

$\text{Previous_x} = x2;$

$\text{Previous_y} = y2;$

Step5:

For all $v \in \text{neighbors}[u]$

Do if $d[v] > d[u] + w[u, v]$ //if new shortest path found.

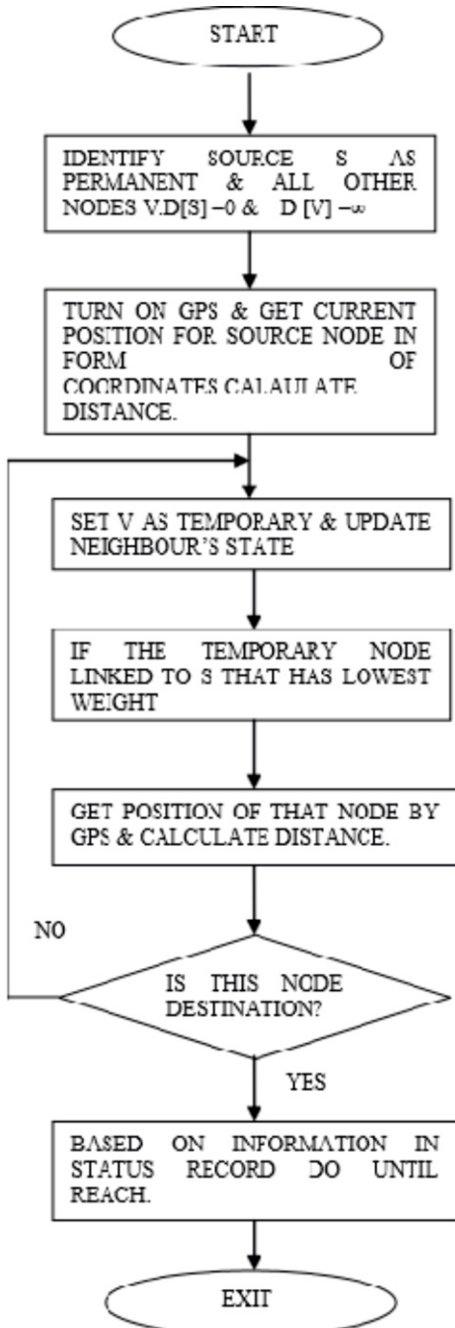
Then

$d[v] = d[u] + w[u, v]$ //see new value of

shortest path if desired then trackback.

Return dis.

Step6:
 Get the position (x2, y2) of the visited nodes from GPS
 Current_x = x2;
 Current_y = y2;
 $distance = \sqrt{(x2 - source\ x)^2 + (y2 - source\ y)^2}$
 dist = distance + dist
 Previous_x = x2;
 Previous_y = y2;



III. PROPOSED SYSTEM

A. Registration and operation:

We will be developing application for ANDROID OS for which we will use the Android SDK software and we'll configure server using ASP/JSP. The employees will register using the developed application.

B. Creating Ride

Step 1: Car owner will enter the source, destination, starting time and available seats as input to the android application.

Step 2: This ride creation request will be transferred to the carpool server.

Step 3: Now server will check for existence of route between entered source and destination and will validate the other input information.

Step 4: Now ride is created and ride seeker are able to search and browse this ride.

C. Browsing ride

Step 1: Ride seeker will enter the source, destination, starting time as input to the android application to search and browse for rides.

Step 2: This ride browse information will be transferred to carpool server.

Step 3: Now server will validate all inputs specified by the user.

Step 4: After validation the server will apply the Route Matching Algorithm and show available rides to the ride seeker. Ride seeker can send a request to anyone of these ride creators.

CONCLUSION

Carpooling system is very effective means to reduce pollution and the congestion of vehicles in cities. It provides an eco-friendly way to travel as well as an opportunity to meet new people. Nowadays most people prefer personal vehicle to travel due to delay caused in public transport system and luxuries provided by private vehicles. Pre-registration ensures that only identified people get into the vehicle so that trust can be established. It also provides an opportunity to meet new people. The people registered are allotted specific days on which they should take their private vehicle, so that no inconvenience is caused to its registered passengers for daily commute.

REFERENCES

1. "Real time carpooling system", N.V.Pukhovskiv, R.E.Lepshokov, Ostfold University College.
2. Miguel A. Vargas, Jose I. Walteros, Andres L.Medaglia, 'Car Pooling Optimization: A case Study in Strasbourg(France)', Proceedings of the 2008 IEEE Systems and Information Engineering Design Symposium, University of Virginia, Charlottesville, VA, USA, April 25, 2008.
3. "Implementation of GPS Enabled Carpooling System", SmitaRukhande, Prachi G, Archana S, Dipa D, Dept. Of Information Tech., Mumbai University, Navi Mumbai City, Maharashtra, India, International Journal of Advances in Engineering & Technology, Nov 2011