ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 6, June 2016

International Journal of Advance Research in
Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

An implementation of Genetic Algorithm approach to solve carpool service problem using cloud computing

Teja Shinde[1]
Computer Science
SRCOE
Pune, India

Baban Thombre[2]
Computer Science
SRCOE
Pune, India

Abstract: There has been rapid industrialization and urbanization in the recent past. Due to this increasing pace, a lot of problems like traffic congestion have come to the fore. The harmful effects of traffic congestion are not unfamiliar to many. To ease and mitigate this traffic congestion issue, Carpooling has come out as a decent solution. Carpooling helps to increase the occupancy rates of the car, enabling extra users to be accommodated in the same vehicle, and thus contributing towards the traffic congestion issue. This paper proposes an Intelligent Carpool System called ICS and also describes the costing charges applied on the passengers enjoying carpooling services. According to this system, drivers and passengers can access the Carpool services exercised by the carpool agency through their simple handheld devices. The driver and passenger matching is done via the Genetic carpool route and matching Algorithm called GCRMA. The costs mentioned above which will be charged for the passenger will be calculated via the Haversine Formula.

Keywords: Carpool service problem (CSP), genetic algorithm, intelligent carpool system (ICS).

I. Introduction

There has been tremendous increase in the number of cars on the roads now a days. This increasing number of cars have been attributed to the rapid rates of industrialization and urbanization. Carpooling serves as a feasible solution for mitigating this problem and reducing the number of cars on the road. Through carpooling, the drivers who travel in vehicles which have fewer people can allow additional passengers who have the same destination as the driver to hop on to their vehicles. This allows for increasing the occupancy rate for the cars. Thus the same amount of people can travel in fewer vehicles leading to a substantial reduction in the number of cars on the road.

In addition to the benefit mentioned above carpooling also allows for reduced vehicle emissions, reduced travelling costs and produce a greener environment in turn.

There are many conventional carpooling websites called as Intelligent Carpooling Systems (ICS). These systems include sending requests to the carpooling agency at a specific date and time. These requests coming via drivers and passengers are then matched and a result showing a pick up and drop for the driver and the passenger will be shown. The problem of solving these driver passenger matching location based scenario is called as the Carpool Service Problem or CSP.

II. Related Work

Work on CSP has been undertaken through 2 approaches. They are mentioned briefly in the following paragraphs:

One of the method used was of integer programming. In this approach the problem of CSP was tried to be solved using integer programming. However, integer programming belongs to the family of exact optimization and as such produces similar outputs in different runs of the same problem. Hence, it did not prove to be an effective method to solve CSP.

Another method proposed to solve this problem was using the genetic algorithm approach. Here, the design of the recombination is implemented through a single point crossover operator. As such it has five operations for its mutation procedure. Namely, a push backward, push forward, a remove insert, and a transfer operator ending the process. All of the above operations are performed in a random manner. To garner a deterministic solution to a problem, random algorithms don’t work well. Also, the above mentioned genetic algorithm based approach lacks a problem specific orientation to solve the CSP, so doesn’t qualify for a good solution.

III. Implementation Details: Architecture of the System

The architecture of the system is described as above. The system consists of 2 modules: the Mobile Client (MC) module and the CCS (Cloud Carpool Server) Module. As seen from Figure 1, the system has a Client Server based approach. The MC module is built for the mobile operating system Android. It also has built in functionalities for GPS to gather the location information for mobile communication. Using such services the users can easily obtain information about their current locations by automatically getting accessed and connected to the GPS signals of satellites and they are also able to retrieve geo resource images using the Web Map Service (WMS) application programming interface (API). The users can access and enter their pickup and drop off locations using the mobile application. The drivers and passengers use the mobile application to enter their locations and constraints, and also to get information about their pick-ups real time.

The Cloud Carpool Server (CCS) module helps in matching the respective drivers with their passengers. All the information entered by the drivers and passengers is fed here, and the CCS module then comes up with a solution in the form of a mapping table using genetic algorithm.

The costing module introduced here, calculates the cost which will be incurred on the passenger for the amount of distance he travels using the Haversine Formula.

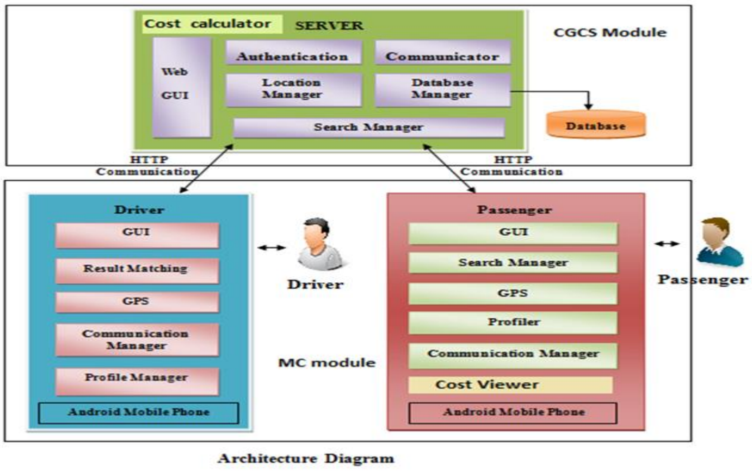

Figure 1: Architecture Diagram

IV. Implementation Details: Genetic Algorithm

Here, Genetic Algorithm is used to find a solution to the CSP. The drivers and the passengers are mapped and a pick point table is displayed as a solution stating which driver and passenger will get together for the journey.

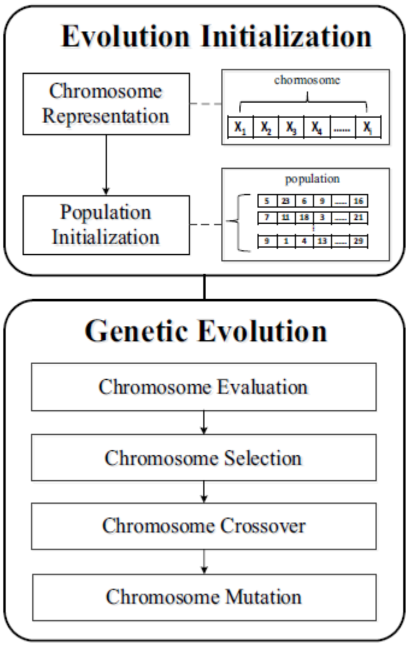

Figure 2: Genetic Algorithm

The existing approach to solve CSP is implemented through GCRMA. The steps for GCRMA are as shown in the figure 2. It consists of Evolution Initialization and Generic Evolution as the two main modules.

Let us look at the different steps in the Evolution Initialization module.

Chromosome Representation:

Each carpool request (Req) is represented by a quintet (RID, RN, RL, RD, RC) that contains several properties, such as the identity number RID, the seat number RN, the current location RL, the destination RD, and the request user category RC: 1 if submitted by a driver or 0 if submitted by a passenger. This entire scenario is shown in the Chromosome Representation phase which is as shown below.

Here, the chromosome representation has an assignment layer wherein every driver is assigned to a passenger and an implicit routing layer where the order in which each driver will pick up a passenger is maintained.

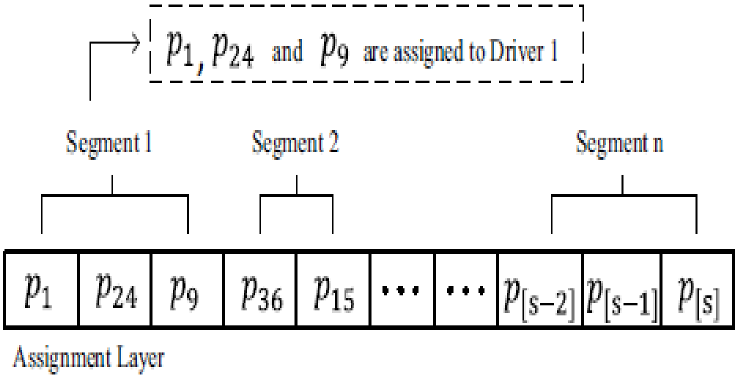

Figure 3: Chromosome Representation

Population initialization.

This steps distributes the initial population in the solution space. Here, it is done in a random manner.

Generic Evaluation Module.

Chromosome Evaluation

This step consists of the fitness evaluation function. All the chromosomes are evaluated via the fitness evaluation function. In this case, the evaluation function checks for the most efficient route that the drivers and the passengers should take. The output of this step will arrange all the chromosomes in the descending order of their function evaluation result. The first 100 chromosomes that is the most elitist chromosomes out of the ones mentioned latter will be selected for the next step of genetic evolution.

Chromosome Crossover

This crossover procedure is utilized to recombine the chromosomes of selected parents to simulate the natural process of evolution.

The new crossover phase as mentioned in the following figure will enable a certain percentage to be taken from the two parents randomly to be combined into a child chromosome having qualities of both the parents. This recombination approach helps to get a higher probability of matching chromosomes and can help in finding an early solution to the problem.

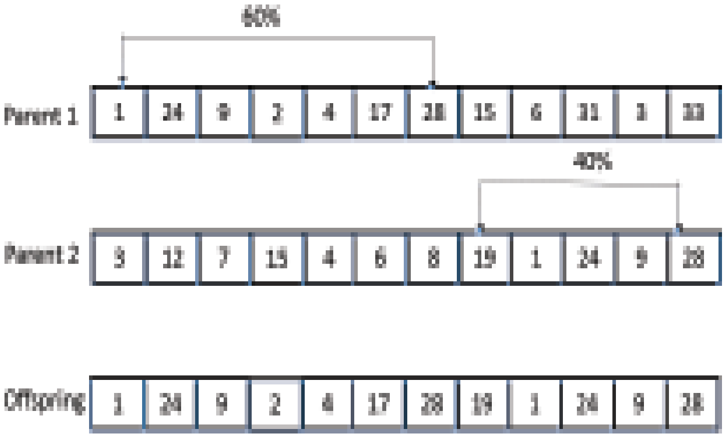

Figure 4 Chromosome Crossover

This step is the final step in genetic algorithm and as such is used only per chromosome. This is used to change the allocation of the passengers within one chromosome using the swap operator. This step is shown in Fig. 5. The pm1 and pm2 of the parents are used to swap with each other and create a new offspring with the resulting swapped operators.

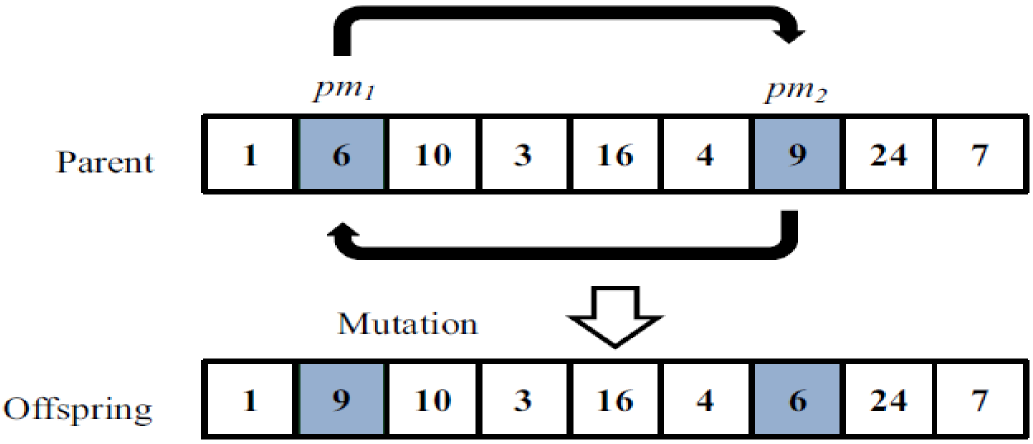

Figure 5: Chromosome Mutation

V. RESULT AND DISCUSSION

The comparative chart of result is given as follows for driver seeker datasets. This chart compares the time and accuracy of the existing system and the proposed system. We are using WEKA tools for this comparison and analysis.

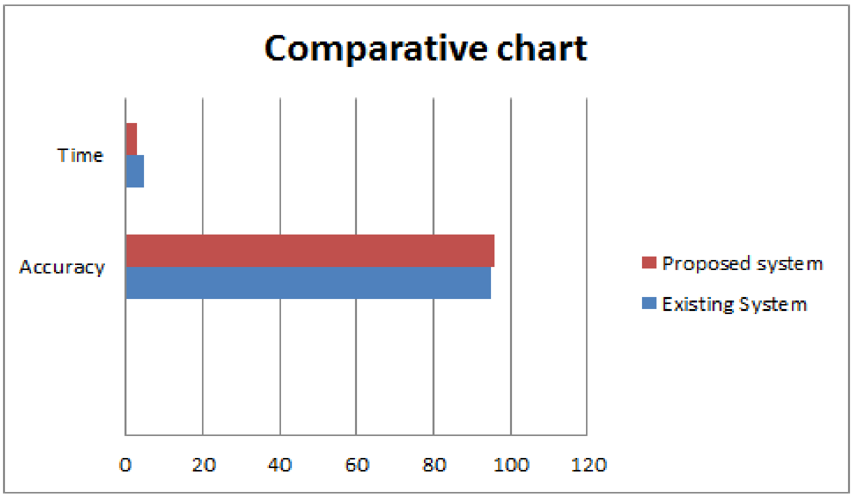

Figure 6: Comparative Chart

VI. CONCLUSION

This paper covers all the aspects of the ICS system which is used to find suitable driver passenger mappings by forwarding requests and receiving responses from the ICS. The entire architecture of the ICS system is discussed. The GCRMA algorithm which is used in the existing system to map the appropriate driver and passenger using genetic algorithm has also been covered. The proposed system adds a costing module to the existing system using Haversine formula to calculate the aerial distance. It also modifies the genetic algorithm such that a large number of permutations takes place to increase the possibility of finding the solution in the crossover phase itself. In addition to this the proposed system also supports the platform independence feature in mobiles using the phonegap technology.

ACKNOWLEDGEMENT

I am profoundly grateful to Baban Thombre for his expert guidance and continuous encouragement throughout to see that this paper rights its target since its commencement to its completion. I would like to express deepest appreciation towards his invaluable guidance and support me in completing this paper. At last I must express my sincere heartfelt gratitude to all the staff members of Computer Department who helped me directly or indirectly during this course

References

1. Shih-Chia Huang, Ming-kai Jiau and Chih-Hsing Lin, “A Genetic Algorithm Based Approach to solve Carpool Service Problems in Cloud Computing,” IEEE Transactions on Intelligent Transportation Systems, 2014
2. S. Hartwig and M. Buchmann, “Empty Seats Traveling,” Nokia Research Center, Bochum, Germany, Feb. 2007.
3. R. Fagin and J. H. Williams, “A fair carpool scheduling algorithm,” IBM J. Res. Develop., vol. 27, no. 2, pp. 133–139, Mar. 1983.
4. R. K. Megalingam, R. N. Nair, V. Radhakrishnan, and A. V. Vidyapeetham,“Automated wireless carpooling system for an eco-friendly travel,” Electron. Comput. Technol., vol. 4, pp. 325–329, Apr. 8–10, 2011.
5. “Carma’s Home Carma’s,” Carma’s Corporation, accessed April 22, 2014.[Online]. Available: https://carmacarpool.com/
6. “Sidecar’s Home,” Sidecar’s Corporation, accessed April 22, 2014.[Online]. Available: http://www.side.cr/
7. “Flinc’s Home,” Flinc’s Corporation, accessed April 22, 2014. [Online].Available: https://flinc.org/
8. “Zimride’s Home,” Zimride’s Corporation, accessed April 22, 2014.[Online]. Available: http://www.zimride.com/
9. “go2gether’s Home,” go2gether’s Corporation, accessed April 22, 2014.[Online]. Available: http://www.go2gether.ca/
10. N. Agatz, A. L. Erera, M. W. P. Savelsbergh, and X. Wang, “Optimization for dynamic ridesharing: A review,” Eur. J. Oper. Res., vol. 223, no. 2,pp. 295–303, Dec. 2012.
11. P. Bianco, R. Kotermanski, and P. Merson, “Evaluating a ServiceOriented Architecture,” Software Engineering Institute, Pittsburgh, PA,USA, CMU/SEI-2007-TR-015, Sep. 2007.