Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

A CAR POOLING MODEL WITH CMGV AND CMGNV STOCHASTIC VEHICLE TRAVEL TIMES

POOVIZHI M[1], Dr. S UMA[2], SINDHUJA P[3], DEEPACHANDRA D[4]

[1] PG Scholar, PG CSE Department, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, India
[2] Head of the Department, PG CSE Department, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu, India

Abstract- Carpooling (also car-sharing, ride-sharing, lift-sharing), is the sharing of car journeys so that more than one person travels in a car. It helps to resolve a variety of problems that continue to plague urban areas, ranging from energy demands and traffic congestion to environmental pollution. Most of the existing method used stochastic disturbances arising from variations in vehicle travel times for carpooling. However it doesn't deal with the unmet demand with uncertain demand of the vehicle for car pooling. To deal with this the proposed system uses Chance constrained formulation/Programming (CCP) approach of the problem with stochastic demand and travel time parameters, under mild assumptions on the distribution of stochastic parameters; and relates it with a robust optimization approach. Since real problem sizes can be large, it could be difficult to find optimal solutions within a reasonable period of time. Therefore solution algorithm using tabu heuristic solution approach is developed to solve the model. Therefore, we constructed a stochastic carpooling model that considers the in- fluence of stochastic travel times. The model is formulated as an integer multiple commodity network flow problem. Since real problem sizes can be large, it could be difficult to find optimal solutions within a reasonable period of time.

Index Terms— Carpooling, multiple commodity network flow problem, stochastic travel time, time–space network.

I. INTRODUCTION

1.1 ANDROID

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as smart phones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it also has been used in game consoles, digital cameras, and other electronics. The model is formulated as a mixed integer linear program whose problem size is expected to be quite large in real applications. An algorithm is developed to efficiently solve the model. Finally, to evaluate how the model performs in practice, a number of cases are randomly generated, based on data from Guo's [15] study of northern Taiwan. The remainder of this paper is organized as follows. The model formulation is first described and then the development of an algorithm to solve the model. Thereafter, a case study is performed to evaluate the performance of the model. Finally, some conclusions are given.

Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices. Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users or bring Android to devices which were officially, released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "smart phone wars" between technology companies.

II. LITERATURE SURVEY

Some advanced local search algorithms have been developed that can directly solve concave cost bipartite network problems. Moreover, the effectiveness of these modified local search algorithms for solving general concave cost transhipment problems is doubtful. In this research, we propose a global search algorithm for solving concave cost transhipment problems. Efficient methods for encoding, generating initial populations, selection, crossover and mutation are proposed, according to the problem characteristics. To evaluate the effectiveness of the proposed global search algorithm, four advanced local search algorithms based on the threshold accepting algorithm, the great deluge algorithm, and the tabu search algorithm, are also developed and are used for comparison purpose. To assist with the comparison of the proposed algorithms, a randomized network generator is designed to produce test problems. All the tests are performed on a personal computer. The results indicate that the proposed global search algorithm is more effective than the four advanced local algorithms, for solving concave cost transhipment problems.Limitations are Difficult to find solution for a large scale problem.

Perturbations of flight schedules may occur everyday. Poor scheduling of flights may result in a substantial loss of profit and decreased levels of service for air carriers. This research aims at developing a framework to help carriers in handling schedule perturbations caused by the breakdown of aircraft. The framework is based on a basic schedule perturbation model constructed as a dynamic network from which several perturbed network models are developed for scheduling following incidents. These network models are formulated as pure network flow problems or network flow problems with side constraints. The former are solved using the network simplex method while the latter are solved using Lagrangian relaxation with sub gradient method.Limitations are Pure network flow problems.

Seak hour travel is a perennial headache for many Americans — peak hour travel time average 200 hours a year in large metropolitan areas — some cities have managed to achieve shorter travel times and actually reduce the peak hour travel times. The key is that some metropolitan areas have land use patterns and transportation systems that enable their residents to take shorter trips and minimize the burden of peak hour travel. That’s not the conclusion promoted by years of highway-oriented transportation research. The Urban Mobility Report (UMR) produced annually by the Texas Transportation Institute and widely used to gauge metropolitan traffic problems has overlooked the role that variations in travel distances play in driving urban transportation problems. This report offers a new view of urban transportation performance. It explores the key role that land use and variations in travel distances play in determining how long Americans spend in peak hour travel. Limitations are spending much more total time.

Most past car pooling studies have focused on the to-work problem (from different origins to a common destination) or the return-from-work problem (from the same origin to different destinations). Pre-matching information, including the carpool partners and the route/schedule for each previously participating vehicle, have rarely been considered. As a result, there has not yet been a suitable method/model developed for solving practical many-to-many car pooling problem with multiple vehicle and person types, as well as pre-matching information, that occur in real-world. we strive to make up this lack by employing a time–space network flow technique to develop a model for this type of car pooling problem with pre-matching information (CPPPMI). The model is formulated as an integer multiple commodity network flow problem.

A solution algorithm, based on Lagrangian relaxation and a heuristic for the upper bound solution, is developed to solve the model. To test how well the model and the solution algorithm may be applied to real-world, numerical tests are performed with several problem instances randomly generated based upon data reported from a past study carried out in northern Taiwan. The test results show the effectiveness of the proposed model and solution algorithm. Limitations are Numerical tests are performed with several problem.

Difficulty of public transport to adequately cover all passenger transportation needs, different innovative mobility services are emerging. Among those are car pooling services, which are based on the idea that sets of car owners having the same travel destination share their vehicles. Until now these systems have had a limited use due to lack of an efficient information processing and communication support. An integrated system for the organization of a car pooling service is presented, using several current Information and Communication Technologies (ICT's) technologies: web, GIS and SMS. The core of the system is an optimization module which solves heuristically the specific routing problem. The system has been tested in a real-life case study.Limitations are lack of an efficient information processing.

III. EXISTING WORK

In existing system, carpooling is done with stochastic disturbances arising from variations in vehicle travel times. It employed network flow techniques and a mathematical programming method to develop a carpooling model and a solution method of constraint relaxation technique is used accounting for stochastic vehicle travel times. Car pooling model generation based on the two things. One is CMGV is a CMG (carpool member group) with a vehicle. In other words, it is with a vehicle paired with a driver who is a member of the CMGV.

CMGNV is a CMG that cannot provide a vehicle. Another one is CMGNV will be determined to be/not to be served via the optimization process. A clustering ant colony technique is used to solve the Long-term Car Pooling Problem. The proposed Clustering Ant Colony algorithm (CAC) is based on the Ant Colony Optimization (ACO) paradigm; the objective of this work is to cluster the users into car pools by the ant making its tour. It give efficient car pooling.

3.1.1 Limitations of the Existing System

The Existing System has the limitations which are related to the knowledge about the different attributes and the other is about the algorithm used in the existing system and the last is that which deals with the performance of the system. Some of the Limitations of the existing system are given below:

- Not efficient for long term carpooling
- Not able to give better solution

IV. PROPOSED WORK

There are two different forms of car pooling problem: the Daily Car Pooling Problem (DCPP) and the Long-term Car Pooling Problem (LTCPP) , proposed method used for solve the long term car pooling problem. The proposed Clustering Ant Colony algorithm (CAC) is based on the Ant

Colony Optimization (ACO) paradigm; the objective of this work is to cluster the users into car pools by the ant making its tour.

Our ultimate goal is to solve the LTCPP efficiently and obtain a good solution with limited exploration to the search space. In order to achieve these objectives, we introduce a preference concept into the ACO system to replace the traditional pheromone information, which converts the classic ACO into a clustering methodology. We will be developing application for ANDROID OS for which we will use the Android SDK software and we'll configure server. The employees will register using the developed application. Vehicle owner will give source and destination (including the actual path), number of seats, vehicle no. and starting time as input to application.

The intermediate location from source to destination which will be available in the server will be available to vehicle owner for selection. The owner will select one of the paths. The server will be configured either in JSP or ASP. Time is a very important constraint in our project. All the updation should happen instantly without fail. Now the other side of application, fellow who want ride will subscribe to the application. During subscription a fellow will enter source and destination.

This request will be sent to the server. The server will respond with vehicle available on that route. Vehicle number, owners name, seats available etc. will be visible to the fellow. Now this fellow will send request to the vehicle owner through the server. The vehicle owner on the other side will approve the request.

Then we can track the vehicle owner's mobile (using getlat, getlog methods) location and provide it to the fellow. The fellow can see the car location on GMAP. An application signature is required for GMAP for which we will have to register. Once all the vacant seats get full then the vehicle owner can ignore the upcoming requests. For the security and privacy reasons the vehicle owner can decide that whether to accept the request or not. Drivers will need a very convenient interface for specifying their destination and their route options. Riders will need a convenient interface for specifying their starting and ending points, and how much flexibility they have in either (for example, they can walk n blocks from current location in order to be picked up).

4.1 MODULE DESCRIPTION

* CMGV vehicle flow networks,
* CMGV passenger-flow networks,
* CMGNV passenger-flow networks
* Clustering ant colony algorithm for LTCPP
* Performance analysis.

4.1.1. CMGV vehicle flow networks

Each CMGV vehicle-flow time–space network represents potential movements for a CMGV and vehicle within a certain time period between certain locations. The horizontal axis represents the location; the vertical axis represents the time duration. "Nodes" and "arcs" are the two major components in the network. The flow unit for the CMGV vehicle-flow network is a "group." Since a CMGV may be assigned as a CMGVP, the node supply/demand is set as a binary variable. This issue will be addressed in more detail in the CMGV passenger-flow time–space networks. There are two types of arcs.

A node stands for a location at a specific time, whereas an arc designates an activity for the vehicles. The arc flows express the flow of the CMGV in the network. The vertical length of the network is set according to the length of the time window specified for the CMGV. The starting time for the time window (when the network begins) is the earliest time at which the CMGV leaves the original location; the ending time for the time window (when the network ends) is the latest time at which the CMGV wishes to arrive at the destination. The length of each CMGV vehicle flow network may be different.

4.1.2. CMGV passenger-flow networks

To facilitate problem solving, these networks are designed to be analogous to the CMGV vehicle-flow time–space networks; the number of network layers and network lengths are the same as those in the CMGV vehicle-flow networks. In addition, side constraints are used to ensure that each CMGV can flow in only one network between the CMGV vehicle-flow and the associated CMGV passenger-flow networks.

The horizontal/vertical axes, the length of the network, and supply/demand nodes are the same as those described for the CMGV vehicle-flow networks. Each CMGV passenger-flow network represents the potential movement of the CMGVP within a certain time period and at certain locations. The CMGV is assigned as driving a car or as a CMGVP during the optimization process.

A CMGVP holding arc indicates the holding of the associated CMGVP at a certain location in a specific time period. The arc cost is the time cost incurred for holding the CMGVP at this location within the corresponding time period. Similar to the design of holding arcs in the CMGV vehicle-flow networks, CMGVPs do not need to arrive before their departure time or stay after their arrival time. Therefore, the CMGVP holding arc cost associated with the corresponding origin and destination location set to zero and the arc flow is indicated by a binary variable. CMGVP stochastic travel arcs are designed to determine a suitable travel time for each CMGVP trip in the planned schedule.

Similar to the CMGV vehicle-flow time–space network, to reflect the cost incurred for adjusting its schedule, which is not the planned one, and its downstream trips' schedules, in the operation stage, we design an unanticipated penalty value for each scenario of every CMGVP. The total unanticipated penalty value for each trip τ is calculated as follows:

PGm,s τ = □ w∈Sτ,w□=s

pw τ ×gm,w,s τ

= w∈Sτ,w□=s

pw τ ×vm ×etm,w,s τ ×λ

\+ w∈Sτ,w□=s

pw τ ×vm ×ltm,w,s τ ×ρ×γτ (2)

where PGm,s τ is the expected unanticipated penalty value for plan- ning the possible travel time s for trip τ in the mth CMGV passenger-flow network; gm,w,s τ is the unanticipated penalty value for scenario w that is inconsistent with the planned scenario s for trip τ in the mth CMGV passenger-flow network; λ is the early arrival penalty per time point for a CMGVP; γτ is the multiplier value for trip τ for a CMGVP, which is designed similar to ατ. Equation (2) shows the calculation of the unanticipated penalty value for a CMGVP stochastic travel arc. Similar to the unanticipated penalty value designed for a CMGV trip, it is divided into two types, i.e., an early arrival penalty and a late arrival penalty.

The transportation time is the same as the corresponding time block for the associated CMGV trip in the corresponding CMGV vehicle-flow network. The arc cost is a variable time cost representing the CMGVP travelling between locations plus an unanticipated penalty. The arc flow is indicated by a binary variable. Similar to the CMGV vehicle-flow time–space network, to reflect the cost incurred for adjusting its schedule, which is not the planned one, and its downstream trips' schedules, in the operation stage, we design an unanticipated penalty value for each scenario of every CMGVP.

4.1.3. CMGNV passenger-flow networks

To facilitate problem solving, these networks are designed to be similar to the CMGV passenger-flow time–space networks. The horizontal and vertical axes are the same as those in the CMGV passenger/vehicle-flow network. However, each CMGNV passenger-flow time–space network associated with a specific combination of origin, destination, time window, group, and vehicle type (called an ODTWGV pair for simplicity) represents a potential movement of CMGNVs within a certain time period and certain locations. Unlike a CMGV that is associated with a given vehicle type, a CMGNV is associated with a specific combination of origin, destination, time window, and group (called an ODTWG pair for simplicity), which could be assigned to different vehicle types.

A CMGNV holding arc indicates the holding of CMGNVs at a certain location in a specific time period. The arc cost is the time cost incurred for holding a CMGNV at this location in the corresponding time period. Similar to the design of holding arcs in the CMGV passenger-flow networks, CMGNVs do not need to arrive early before their departure time or stay after their arrival time. Therefore, the CMGNV holding arc cost is set to zero associated with its corresponding origin and destination locations. The arc flow's upper bound is infinity. The arc flow's lower bound is zero. A CMGNV unserved arc directly connects the supply node to the demand node. The arc cost is set as a large unserved penalty per person multiplied by the number of people in the CMGNV. The arc flow's upper bound is infinity, and the arc flow's lower bound is zero.

4.1.4 Clustering ant colony algorithm for LTCPP

In CAC, to direct the clustering activity of the ant, we introduce a preference concept into the ant system to replace the traditional pheromone information. The preference information is considered as the preference of one user willing to be pooled in the same cluster with another. When an ant starts a tour from a user, it starts to build a car pool in the meantime.

The ant then behaves according to a roulette wheel selection based on the preference values; it can either visit and insert a new user into its current car pool or end the current car pool and select a new user to start a new car pool. When all the users are visited by the ant, the tour of the ant is considered finished. After all ants finish their tour in the current iteration, several solutions with the least cost are selected to be applied a variable neighborhood search.

At last, in the end of iteration, the preference information between the users in the same car pool of each solution will be increased. By this mechanism, the clustering experience is always memorized and updated to direct the ant search of future iterations.

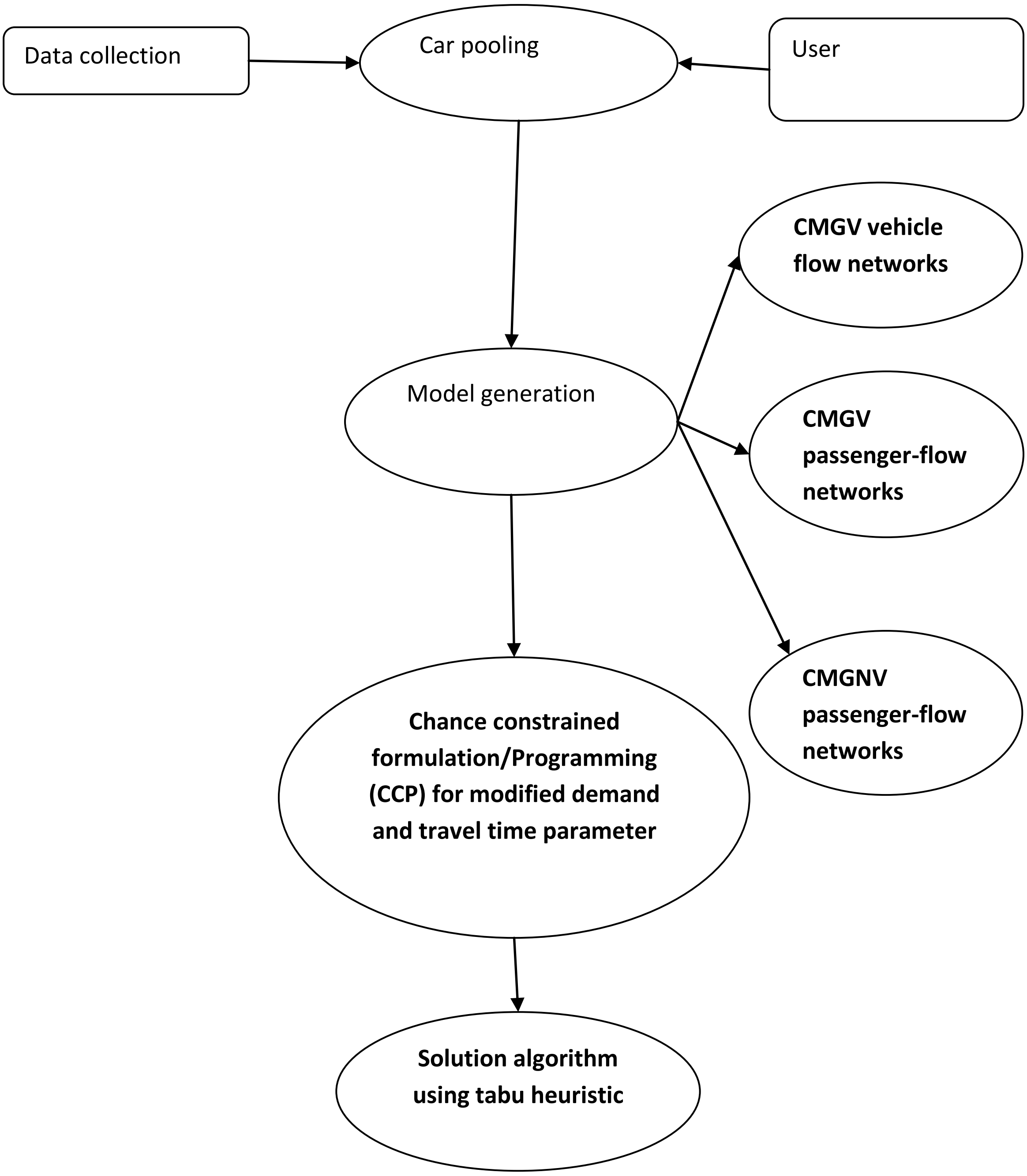

Figure 2.1 System Architecture diagram-level 3

V. CONCLUSION

The increased use of private vehicles has caused significant traffic congestion, noise and air pollution. Public transport is often incapable of effectively servicing non-urban areas. Car pooling, where sets of car owners having the same travel destination share their vehicles, has emerged to be a viable possibility for reducing private vehicle usage around the world**.** The Clustering ant colony algorithm can provide good solutions quality compared with other meta-heuristics. Thus, it has been demonstrated that the CAC algorithm is an effective approach for solving the long-term carpooling problem.

REFERENCES

1. Archetti,C Hertz.A, and Speranza.M, A tabu search algorithm for the split delivery vehicle routing problem.
2. Baldacci.R,Maniezzo.V, and Mingozzi.A, “An exact method for the carpooling problem based on Lagrangean column generation,” *Oper. Res.*,vol. 52, no. 3, pp. 422–439.
3. Calvo.R.W, LuigiF.L, Haastrup.P, and Maniezzo.V, “A distributed geographic information system for the daily car pooling problem,” *Comput. Oper. Res.*, vol. 31, no. 13, pp. 2263–2278.
4. Calvo.R.W and Colorni.A, “An effective and fast heuristic for the diala-ride problem,” *4OR: A Quart. J. Oper. Res.*, vol. 5, no. 1, pp. 61–73.
5. Coslovich.L, Pesenti.R, and Ukovich.W, “A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem,” *Eur.J. Oper. Res.*, vol. 175, no. 3, pp. 1605–1615.
6. Du.Y and Hall.R, “Fleet sizing and empty equipment redistribution for center-terminal transportation networks,” *Manage. Sci.*, vol. 43, no. 2,pp. 145–157.