TOWARDS SOCIAL AWARE RIDESHARING GROUP QUERY SERVICES

Silpa K M[1], Mary Priyanka K S[2]

[1]*Post Graduation Student, Dept. of Computer Science and Engineering, College of Engineering Kidangoor, Kerala, India*
[2]*Assistant Professor, Dept. of Computer Science and Engineering, College of Engineering Kidangoor, Kerala, India*

---***---

Abstract - *Private car ownership in the context of the ongoing urbanization is creating challenges concerning environmental pollution, high energy costs, and limited and expensive parking. As a reaction to these negative impacts, companies are developing new mobility alternatives to private car ownership. One alternative is ridesharing that provides individuals with taxis from a fleet on an as-needed basis. The objective of taxi ride sharing and carpooling is to share the taxi in an efficient way by reducing the time and cost of the passengers, which will also reduce the road traffic. Despite the potential to provide significant societal and environmental benefits, ridesharing has not so far been as popular as expected. Notable barriers include social discomfort and safety concerns when traveling with strangers. To overcome these barriers, this paper propose a new type of Social-aware Ridesharing Group (SaRG) queries which retrieves a group of riders by taking into account their social connections and spatial proximities.*

***Keywords*: Location-based Services, Query Processing, Ridesharing, Group Queries, Social Acquaintance**

1. INTRODUCTION

The world faces human-made hazardous weather events such as heat waves, droughts, floods and wildfires in dimensions which have never been seen before. A crucial contributor to this negative trend is the constantly growing transportation sector. In addition, most urban regions suffer from traffic congestions which lead among others to local emissions, the loss of time and noise pollution. One promising approach to reduce the amount of transport related emissions is ride-sharing. The concept of ride-sharing, i.e. users share their ride when their trips match each other in time and place, is one approach to reduce the number of cars and thereby the negative transportation related effects. In addition, there are benefits for the individual user, as fuel, tolls and vehicle costs are shared. Previous studies, in which simulations based on real travel data are performed, have shown that by ridesharing, the number of cars and the kilometres travelled can be significantly reduced. Even if there are already several ride-sharing services on the market, ride-sharing is still no widely-accepted mean of transportation.

Research to investigate user behaviour and acceptance of ride-sharing is still limited, especially testing the acceptance of ride-sharing in real-life settings. Although ridesharing can provide a wealth of benefits, such as reduced travel costs, congestion, and pollution, a number of challenges have restricted its widespread adoption. In fact, even at a time when improved communication systems provide real-time detailed information that could be used to facilitate ridesharing, the share of work trips that use ridesharing has decreased by almost 10 percent in the past 30 years. Notable barriers include social discomfort and safety concerns when traveling with strangers. To overcome these barriers, in this paper, we propose a new type of Social-aware Ridesharing Group (SaRG) queries which retrieves a group of riders by taking into account their social connections and spatial proximities. Because static variety carpool still represents the majority of existing solutions, almost all of the available papers and literature on carpool and ride-sharing mainly tackle the static ridesharing issues, whereby users must pre-schedule their trips, neglecting the dynamic aspects. Despite much of the progress experimented on dynamic carpooling and ridesharing concepts thanks to the current solutions, it still remains in the early stages regarding publicly available works and literature that deal with its real-time automation. In order to make up for that shortfall, some of the papers which mention carpooling and ride-sharing, and even some that did considered the dynamic aspect [8], in majority also considered other issues beside the static and dynamic carpooling and ride-sharing problems at the same time. Some papers are especially involved in the concepts of traceability, communication and security services, which their authors feel that none of the current solutions evoked, identifying the security issues as one of the main reasons hindering their success.

Ride-sharing is an effective way to reduce the number of cars on the streets in order to address both individual and city-wide issues. On one hand, individuals are interested in reducing the cost of their car usage and save on gasoline and other usage-based costs [2]. On the other hand, cities are interested in reducing traffic and pollution and provide incentives to encourage commuters to share rides. As said, the expenses, both environmental and fiscal, of single occupancy vehicles could be reduced by utilizing the empty seats in personal transportation vehicles. Carpooling and ride-sharing target those empty seats: taking additional vehicles off the road reducing traffic and pollution, whilst providing opportunities for social interaction. However, historically carpool scheduling often limited users to consistent schedules and fixed rider groups–carpooling to the same place at the same time with a set person or a group of people. To make that problem worse, the leading problem concerns, given in a 2009 survey about why people don't carpool, were difficulty to organize carpools and inconvenience of organization [4]. We feel both of those can be addressed by employing some novel web technologies and modern day available data stores which hold social and location based individual user's data.

Because static variety carpool still represents the majority of existing solutions, almost all of the available papers and literature on carpool and ride-sharing mainly tackle the static ridesharing issues, whereby users must pre-schedule their trips, neglecting the dynamic aspects. Despite much of the progress experimented on dynamic carpooling and ridesharing concepts thanks to the current solutions, it still remains in the early stages regarding publicly available works and literature that deal with its real-time automation. In order to make up for that shortfall, some of the papers which mention carpooling and ride-sharing, and even some that did considered the dynamic aspect [13], in majority also considered other issues beside the static and dynamic carpooling and ride-sharing problems at the same time. Some papers are especially involved in the concepts of traceability, communication and security services, which their authors feel that none of the current solutions evoked, identifying the security issues as one of the main reasons hindering their success.

Social networks represent certain types of social interaction, such as acquaintance, friendship, or collaboration between people or groups of people that are referred to as actors [7]. In social networks, vertices (nodes, dots) usually stand for actors, and edges (arcs, links, lines) represent the pair wise relations or interactions between the actors. One of the central concepts in social network analysis is the notion of a cohesive subgroup, which is a tightly knit subgroup of actors in a social network. While the notion of a clique embodies a perfect cohesive group, in which every two entities are connected to each other, this definition is overly conservative in many practical scenarios. Indeed,(i) not need to require every possible link to exist between elements of a cohesive subgroup; (ii) the social network of interest may be built based on empirical data, which are prone to errors, so, even if a completely connected cohesive subgroup is sought for, it may be impossible to detect due to erroneous data [7]. To overcome this impracticality of the clique model, other graph-theoretic formalizations of the cohesive subgroup concept have been proposed in the literature.

This model aims to find a ridesharing group with a desired level of social acquaintance. To model such social acquaintance, we assume the existence of a social network graph in which users are connected if they have acquaintance relationships. Such a network might be derived from call graphs based on telephone call detail records (CDRs) or online social networks such as Facebook and Twitter. There are a number of social models that can be employed to measure the social cohesiveness of a ridesharing group, such as star (friend) (one central user has direct connections to all other users), star (friend of friend), and k-core. A k-core of a graph is a maximal connected subgraph in which every vertex is connected to at least k vertices in the subgraph.

Along with social acquaintance, trip matching of a ridesharing group must also be measured. The primary cost of a rider is the travel cost between the riders origin, destination and the drivers origin, destination. The returned ridesharing group should have the smallest travel cost because naturally only riders whose origin and destination are close to those of the driver are willing to join the drivers ridesharing. The minimum travel cost requirement and the social constraint are equally important in our problem [8].

Many real-world graphs are highly dynamic. In social networks, users join/leave and connections are created/severed on a regular basis. In the web graph, new links are established and severed as a natural result of content update and creation. In customer call graphs, new edges are added as people extend their list of contacts. Furthermore, many applications require analyzing such graphs over a time window, as newly forming relationships may be more important than the old ones. For instance, in customer call graphs, the historic calls are not too relevant for churn detection. Looking at a time window naturally brings removals as key operations like insertions. This is because as edges slide out of the time window, they have to be removed from the graph of interest. In summary, dynamic graphs where edges are added and removed continuously are common in practice and represent an important use case.

Ridesharing group query processing is a very recent research topic. The existing studies on this topic fall into three categories: static ridesharing, dynamic ridesharing and trust-conscious ridesharing.

Static Ridesharing : Most of the early studies considered static ridesharing, which refers to the scenario where the requests of drivers and riders are known in advance. We survey static ridesharing in the following categories: slugging, carpooling, and dial-a-ride. Slugging [4] is a typical form of ridesharing where passengers walk to the origin of the driver's trip, board at the departure time, debark at the driver's destination and then walk to their own destinations. Ma and Wolfson [9] studied slugging from a computational perspective using a graph abstraction. Carpooling is another representative application of ridesharing for daily commutes, where private car drivers declare their availability for pick-up and later bring back riders. The main issue in carpooling is the assignment of riders to drivers and the identification of each driver's route to minimize the travel cost. For small-size carpooling, it can be solved by using linear programming techniques [6], [7]. To deal with the large-size problem, several heuristic algorithms have been proposed.

Dynamic Ridesharing : Enabled by recent mobile technologies, dynamic ridesharing services have been gaining increasing attention. In dynamic ridesharing systems, riders and drivers continuously enter and leave the system; dynamic ridesharing algorithms match up them in real time or on a short notice. Existing works can be broadly classified into two categories: centralized and distributed. Centralized real-time ridesharing relies on a central service provider to perform all operations for ridesharing. A recent survey on the optimization techniques for centralized dynamic ridesharing can be found in [2]. Various optimization objectives (e.g., minimizing system-wide vehicle miles or travel time) and spatial-temporal constraints (with desired departure/ arrival time or spatial proximity requirements) have been considered. Rigby et al. [6] proposed an opportunistic user interface to support centralized ridesharing planning while preserving location privacy. The latest work [11] modeled a centralized real-time ridesharing problem with service guarantee, and several novel kinetic tree-based algorithms were proposed to better suit dynamic request scheduling and on-the-fly route adjustment.

Trust-Conscious Ridesharing : A few recent existing works have been made to address the trust issue in ridesharing [1]. Suggested approaches include the adoption of reputation-based systems and profile checking by linking with social networks like Facebook [1]. This approach entails significant involvement from participants. Cici et al. [8] suggested grouping participants who are friends or friends of friends in the assessment of the potential benefits of ridesharing. However, as indicated by our user study result, such simple social constraints would be either too restricted or too relaxed to be practical for real-life ridesharing systems.

2. RELATED WORK

2.1 En-Route Ride-Sharing

Blerim Cici, Athina Markopoulou, Enrique Frias-Martinez, Nikolaos Laoutaris[1], This assesses the potential of ride-sharing for reducing traffic in a city based on mobility data extracted from 3G Call Description Records(CDRs).CDRs are generated when a cellphone makes or receives a call or uses a service, e.g. SMS [9]. Information regarding the time/date and the location of the Base Transceiver Station (BTS), used for the communication, are then recorded. More specifically, the main fields of each CDR entry are the following: (1) the originating cellphone number (2) the destination cellphone number (3) a time-stamp (4) the duration of the call and (5) the BTS tower used by one, or both if applicable, cellphones

First, we infer home/work location of individual users, by adapting state-of-the-art techniques [5] to our CDRs and geotagged tweets. Also, we infer social ties among the users; we use phone calls in the CDR data and explicitly stated friendship in the Twitter data. These ties are later used for social filtering, to address concerns about riding with strangers. Second, given a set of users with known home/work locations, we develop a framework for matching users that could share a ride. Our goal is to minimize the total numbers of cars and provide rides to as many users as possible. We consider several constraints including: spatial (ridesharing with neighbors, i.e. someone within a certain distance from their home/work location), temporal (ride-sharing within a time window from the desired departure/arrival time) and social (ride-sharing with friends or friend-or-friends) constraints. We also consider two versions of the problem:End-Points RS, ride-sharing between home and work locations, and En-Route RS, allowing the possibility to pick up passengers along this route. Third, we use our framework to assess the inherent potential of ride-sharing to exploit the overlap in people's commute in a city. We find that there is significant potential for reducing traffic via ride–sharing, the exact magnitude of which depends on the constraints assumed for matching, as well as on the characteristics of the cities and the type of data set (CDR vs Twitter).

The recorded cell towers of a user are clustered to produce the list of places that the user visits. Then, regression analysis is applied to determine the features of the clusters that represent important places that the user visits. The used features are: (1) the number of days that the user appeared on the cluster; (2) the duration of user appearances on the cluster; and (3) the rank of the cluster based on number of days appeared. Once important locations have been inferred, and the algorithm chooses which of these are home and which are work locations. According to their results, the best features that characterize home and work are: (4) the number of phone calls between 7PM - 7AM, i.e. Home Hour Events, and (5) number of phone calls between 1PM - 5PM, i.e. Work Hour Events.

2.2 Simulation Algorithm

Masayo Ota, Huy Vo, Cl_audio Silva, Juliana Freire [2], STaRS supports the simulation of real-time ride sharing which serves unplanned trips. A wide deployment of ride sharing requires a better understanding of its tradeoffs. This is challenging since there are multiple stakeholders with different, and sometimes conflicting, interests. Governments want less traffic and pollution; taxi companies want to maximize their profits; and passengers would like to reach their destination quickly and cheaply. To design an effective policy, these interests need to be considered [12].

The main components of simulation model are:

Taxi Fleet: The taxi fleet refers to the set of taxis that are involved in the simulation. In contrast to previous works, where taxis are considered as homogeneous objects, to support a multi-vendor environment and different types of vehicles, we consider each taxi as a distinct object with its own specifications.

Passengers: We assume that passengers ride in groups of size greater than or equal to one. Each group is associated with a drop-off location and a set of ride-sharing constraints. **Scheduler:** For each pick-up request, the scheduler finds the most appropriate taxi based on pre-defined metrics. To do so, the scheduler must know all taxi locations along with their current states at all times.

Road Network: The underlying road network of a city is represented as a directed graph. All taxis travel along this road network. Each directed edge represents a road segment, and each node represents the intersection of two or more roads. When a road allows traffic flow in both directions, there are two directed edges for that road.

In the taxi ride-sharing problem, the goal is to minimize the total cost or maximize the total utility of sharing while meeting a set of constraints. A straightforward way to compute the additional cost is to explicitly find an optimal route for the cab that includes the pick-up and drop-off locations of rider and to compare its cost with the cost of the current route for the cab. However, computing the optimal path is known as the Sequential Ordering Problem (SOP) which is a version of the Traveling Salesman Problem and is NP-hard.

2.3 Clique relaxation

Jeffrey Pattillo, Nataly Youssef, and Sergiy Butenko[3], Social networks represent certain types of social interaction, such as acquaintance, friendship, or collaboration between people or groups of people that are referred to as actors. In social networks, vertices usually stand for actors, and edges represent the pairwise relations or interactions between the actors. One of the central concepts in social network analysis is the notion of a cohesive subgroup , which is a tightly knit subgroup of actors in a social network. While the notion of a clique embodies a perfect cohesive group, in which every two entities are connected to each other. This definition is overly conservative in many practical scenarios[7].

To overcome this impracticality of the clique model, other graph-theoretic formalizations of the cohesive subgroup concept have been proposed in the literature. Not surprisingly, all these alternative definitions can be viewed as clique generalizations, each of which relaxes one of the elementary clique properties, such as familiarity, reachability, or robustness. Hence, we use the term clique relaxations in reference to such models.

Relaxation Models

We are considering a simple undirected graph, G=(V,E). A graph is said to be complete if there exists an edge between every pair of vertices in the graph. An induced subgraph is a subset of the vertices of a graph G together with any edges whose endpoints are both in this subset. A clique in a graph is an induced subgraph which is complete.

2.3.1 Relaxation of Reachability

The length of the shortest path between any two vertices u,v in a graph is denoted by d(u,v).A k-clique S is the subgraph of a graph : for all vertices u,v in S, d(u,v)<=k.

2.3.2 Relaxation of Familiarity

Relax the minimum number of neighbors or the maximum number of non-neighbors within the group. Familiarity is another important property one wants to have in a cohesive subgroup. Every member of the group should be familiar with every other member of the group. The k-core concept imposes a lower bound on the minimum degree within the subgraph. Ensures that each vertex in the group is connected to at least k other vertices in the graph. A k-core S is a subgraph of a graph: for every vertices u,v in S, deg(v)>=k [14].

2.4 Demand Responsive Transit (DRT)

Kota Tsubouchi, Kazuo Hiekata, Hiroyuki Yamato[4], On-Demand Bus is a Demand Responsive Transit (DRT) service. It allows potential passengers to request service via the Internet or mobile phone. The requests compose of pick-up location, delivery location and desired delivery time (or pick-up time). The computer executes two main algorithms which are vehicle-choosing algorithm and routing algorithm. After calculation, the system will report to the customer whether the request is accepted or not. If it is accepted, the vehicle will pick up and deliver him to his destination within a guaranteed time.

In this problem, N customers have to be transported by maximum V vehicles. Each customer, customer n, has to specify pick-up bus stop, p(+n), and delivery bus stop, p(-n). He also has to specify either desired pick-up time or a desired delivery time.

Vehicle Choosing Algorithm

We try to introduce an effective algorithm with less calculation time, especially when solving big problems. First, we define direction vector of customer n. On the other hand, we define bus direction vector. Then we define the direction decision variable which is the cosine of the angle between these two vectors. When a new reservation comes into the system, the vehicle-choosing algorithm will be executed for each available bus. Since the bus with the most cosine value is the one with the closest direction to the new demand, that bus will be firstly selected to be executed by the next algorithm, routing algorithm.

Routing Algorithm

For routing algorithm, we developed a heuristic algorithm. In this example, there are *n-1* passengers who have already reserved the bus. Then there is a new reservation from customer *n*. We proposed the "Insertion and time adjustment algorithm", which will insert event p(+n) and p(-n) into the planned route. After insertion, some passengers' pick up time or delivery time will be changed within their each time windows.

2.5 T-Finder

Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, Xing Xie[5], Recommender system for both taxi drivers and the riders. Done by using the knowledge of passengers' mobility patterns and drivers' picking up/dropping off behaviours. First the recommender system provides the drivers with some locations and the routes to these locations . These are the locations towards which they are more likely to find passengers quickly. Second it recommends the people with some locations where they can easily find taxis. Taxis report their present location to a data center in a certain frequency. Besides geo-position and time, occupancy is also recorded. A large number of GPS trajectories are thus generated everyday. These trajectories include two aspects of knowledge:

Passengers' mobility ie, when and where passengers get on and off a taxi. The other are taxis pick up/drop off behaviors. We consider three states for a working taxi: occupied (O), cruising (C) and parked (P) [7]. A taxi trajectory is a sequence of GPS points logged for a working taxi. Each point p has the fields: time stamp p.t, latitude p.lat, longitude p.lon, located road segment p.r, state p.s. A taxi trip is a sub-trajectory which has a single state, either cruising or occupied. We develop an approach to detect the parking places from GPS trajectories and segment the GPS trajectories. We first keep checking the distance between the current point and the latter point until the distance is smaller than a threshold. A large number of GPS trajectories are thus generated everyday which is difficult to handle. Passengers' privacy and their social acquaintance is not concerned. It is based on trip matching and social acquaintance.

3. PROPOSED FRAMEWORK

Ridesharing system consists of three parties: (i) riders (or passengers who want to participate in ridesharing), (ii) drivers (or private car owners who offer ridesharing), and (iii) ridesharing service provider (RSP) (the server in charge of the arrangement of ridesharing). The riders submit ride requests to the RSP, while the drivers send in ride offers.

3.1 Problem Statement

Define an SaRG query over a set of riders D and a social network G=(V,E). Each rider v in D has a ridesharing trip request denoted by tp_v = (o,d) where o and d represent the origin and destination of v's trip, respectively. For the social network G, each vertex v in V is a user and each edge e in E denotes an acquainted relation between two users it connects. Each driver u's ride offer forms an SaRG query q_u . Once the RSP receives an SaRG query q_u from a driver u, it will return u with the most suitable riders from D by considering trip matching and social acquaintance.

3.2 System Architecture

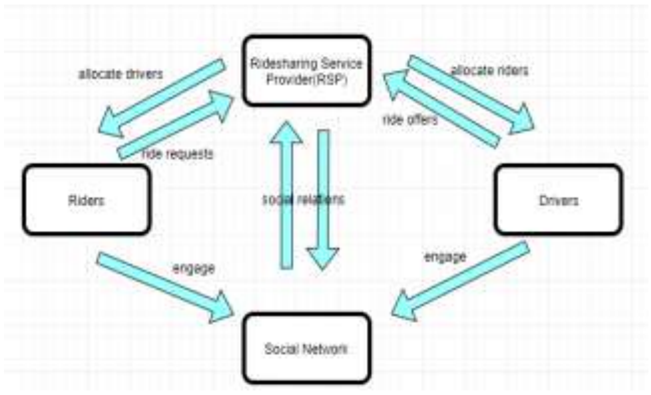


Fig -1: System Architecture

The riders submit ride requests to the RSP, while the drivers send in ride offers. In other words, a ride offer provided by a driver forms an SaRG query; the riders who submitted ride requests form the data space (or search space); the RSP arranges the best ride matches of ridesharing by jointly considering trip matching as well as social connections. In the ridesharing system, we adopt a simple yet popular form of ridesharing called Slugging. Slugging assumes that the driver's trip is fixed and that the riders would walk to the origin location of the driver's trip, board at the departure time, alight at the driver's destination, and then walk to their own destinations.

We define an SaRG query over a set of riders D and a social network G=(V,E). Each rider v has a ridesharing trip request denoted by tp_v=(o, d) where o and d represent the origin and destination of v's trip, respectively. Each driver u's ride offer forms an SaRG query q_u. An SaRG query aims to find a ridesharing group with a desired level of social acquaintance. To model such social acquaintance, we assume the existence of a social network graph in which users are connected if they have acquaintance relationships.

Once there comes a ride offer from a driver, the RSP will match the most suitable riders to the driver. A ridesharing group is composed of a driver and the most suitable riders. For matching the suitable riders to a driver RSP uses a tree like structure which stores the social information of the users which is obtained from the social network site. In the social network site the relationships among users are stored in the form of a tree-like structure. To form a group of riders for particular driver, RSP first checks their spatial proximities and for checking their social connections RSP will perform Deapth First Traversal(DFS) up to a certain level o the tree ad fids the group of riders with a desired level of social acquaintance relationships.

Slugging assumes that the driver's trip is fixed. Riders would walk to the origin location of the drivers trip. Board at the departure time, alight at the driver's destination, and then walk to their own destinations. An SaRG query aims to find a ridesharing group with a desired level of trip matching and social acquaintance. There are a number of social models that can be employed to measure the social acquaintance of a ridesharing group.

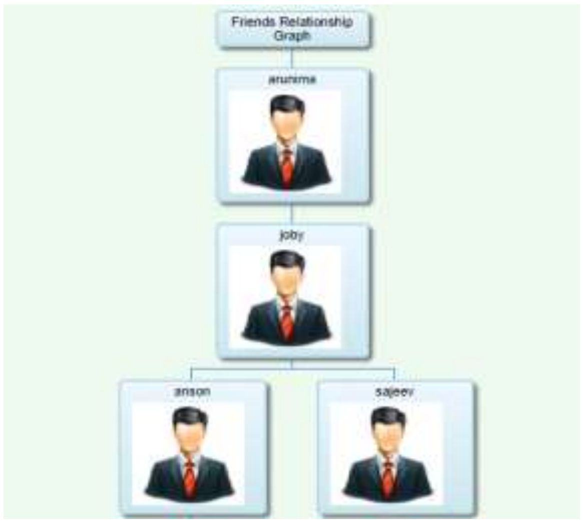
Star (friend): one central user has direct connections to all other users.

Star (friend of friend): one central user has direct or through-a-friend connections to all other users.

k-core: A k-core S is a subgraph of a graph : for every vertices v in S, deg(v)>=k.

The primary cost of a rider in Slugging is the travel cost between the rider's origin, destination and the driver's origin, destination[12]. A ridesharing group consists of a driver u and a size-s set of riders. The size of a ridesharing group is s+1.The returned ridesharing group should have the smallest travel cost. The minimum travel cost requirement and the social constraint are equally important in our problem. This project propose a dynamic and trust-conscious ridesharing system where riders and drivers continuously enter and leave the system and are matched up in real time or on a short notice and which addresses the trust issue in ridesharing.

This system aims to find a ridesharing group with a desired level of social acquaintance. To model such social acquaintance, we assume the existence of a social network graph in which users are connected if they have acquaintance relationships (e.g., friends or colleagues). Such a network might be derived from call graphs based on telephone call detail records (CDRs) or online social networks such as Facebook and Twitter. There are a number of social models that can be employed to measure the social cohesiveness of a ridesharing group, such as star (friend) (one central user has direct connections to all other users), star (friend of friend) (one central user has direct or through-a-friend connections to all other users), and k-core (each user has direct connections with at least k other users). To explain the k-core concept of our system we have to first explain the concept of clique in a graph. A clique C is a subset of vertices in a graph G such that the subgraph G[C] induced by C on G is complete. A clique is called maximal if it is not contained in a larger clique, and it is called maximum if there is no larger clique in G.



This is the tree-like structure of the social relationship obtained from the social network site. Each node in the tree represents each friend of a particular user. If the user receives a new friend request and if he accepts the request, then a new node will be added to the tree dynamically indicating the new friend. RSP performs Deapth First Search (DFS) traversal up to a certain level on the tree like structure and finds the most suitable set of riders for sharing the ride with a desired level of social acquaintance relationship. Level up to which the DFS has to be performed shows the importance that a user gives to the social concerns.

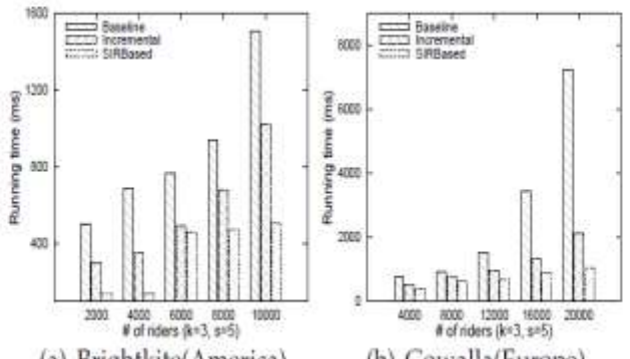
We formulate a new type of Social-aware Ridesharing Group (SaRG) queries to accommodate the real-world need of considering social comfort and trust in ridesharing. An SaRG query retrieves a ridesharing group where each riders trip is similar to that of the driver, and each member of the ridesharing group should be familiar with at least k other members.

We propose an efficient algorithm and a set of efficient pruning techniques to answer SaRG queries. We also devise several incremental strategies by reducing repeated computations to speed up query processing. We also design a novel index structure, Social- Info R-tree (SIR-tree), which integrates social information into R-tree, to further prune the search space and then propose the SIRBased algorithm that integrates the algorithm with the SIR-tree structure. We conduct extensive experiments to evaluate the query efficiency of our proposed algorithms. The consideration of social factors in ridesharing brings several new research challenges. First, how to capture and model social constraints for the purpose of ridesharing is a fundamental issue. Second, the social relationship may not be incremental in nature. As such, the social-aware ridesharing problem becomes more challenging. Indeed, as we shall prove later, the SaRG query problem in this paper is NP-hard, and therefore how to design an efficient algorithm to retrieve the optimal answer to an SaRG query is the focus of this paper. Our key insight is that in practical settings an SaRG query possesses some intrinsic properties.

4. EXPERIMENTAL RESULTS

We evaluate the query processing performance of three algorithms, which are Baseline, Incremental and SIR-Tree, under different parameter settings. Following many other query processing performance evaluation methods, we report the overall query performance in terms of the average elapsed time. We can observe that both Incremental and SIRBased perform better than Baseline. Note that the y-axis is in log-scale. Under different values of s, SIRBased achieves the best performance. This conforms to our theoretical analysis: the SIR-tree structure can efficiently prune many irrelevant users who cannot satisfy either the social diameter or core number constraints as early as possible, leading to a much smaller search space. Even when s is small, SIRBased algorithm performs the best because a small group size leads to a small diameter which results in a good pruning ability of the SIR-tree.

We evaluate the query processing performance of these three algorithms under different parameter settings. Following many other query processing performance evaluation methods, we report the overall query performance in terms of the average elapsed time.



(a) Brightkite(America) (b) Gowalla(Europe)

Effect of the number of riders. In this set of experiments, we show the performance of the algorithms under various numbers of riders (i.e., the size of the rider space) in Figure above. We randomly extract several subsets of the rider space to evaluate the algorithms' performance. As expected, the result demonstrates that SIRBased achieves the best query efficiency in all cases. Compared to Incremental and SIRBased, Baseline is more sensitive to the number of riders. Its query processing time increases rapidly with the increase of the number of riders.

5. CONCLUSION

Introduced a new practical type of SaRG queries that solves the ridesharing problem with flexible social constraints. An SaRG query aims to find a group of riders in which each rider's ridesharing trip is close to that of the query issuer and each member in this group is familiar with at least k other members. We have proposed a series of efficient algorithms to tackle SaRG queries. An extensive empirical study on real datasets demonstrates that the proposed techniques achieve desirable query performance.

Future Enhancement:

First extension is to investigate weighted relations in our social-aware ridesharing group queries. Second, we plan to design more personalized ride requests in our ridesharing system to make our proposed SaRG queries more practical. Third, in some cases, we do not need an exact solution. How to design an efficient approximation algorithm with a tight approximation bound is also our future work.

REFERENCES

1. Blerim Cici, Athina Markopoulou Enrique Frias-Martinez, Nikolaos Laoutaris. Assessing the Potential of Ride-Sharing Using Mobile and Social Data: A Tale of Four Cities. UbiComp '14, September 13 - 17, 2014.
2. Masayo Ota, Huy Vo, Cl_audio Silva, Fellow, IEEE, and Juliana Freire, Member, IEEE. STaRS: Simulating Taxi Ride Sharing at Scale. Ieee transactions on big data, vol. 3, no. 3, july-september 2017.
3. Jeffrey Pattillo, Nataly Youssef, and Sergiy Butenko Clique relaxation models in social network Analysis *Journal of Mathematical Sociology*, 6:139–154, 1978.
4. Kota Tsubouchi, Kazuo Hiekata, Hiroyuki Yamato, Scheduling Algorithm for On demand Bus System, Germany: Springer, 2013.
5. Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, Xing Xie. T-Finder: A Recommender System for Finding Passengers and Vacant Taxis. Parallel Computing, 30(3):377–387, 2004.
6. Yuanyuan He, Jianbing Ni, Xinyu Wang. Privacy-preserving Partner Selection for Ride-sharing Services, Ieee transactions on vehicular technology, vol. 4, 2018
7. Nianbo Liu, Yong Feng, Feng Wang, Bang Liu, and Jinchuan Tang. Mobility Crowdsourcing: Toward Zero-Effort Carpooling on Individual Smartphone. Journal of Operations Research, 59(1):133–142, 2011.
8. Tian, Charles Huang, Yan Liu, Zhi Bastani, Favyen Jin, Ruoming. Noah: A dynamic ridesharing system. Annals of Operations Research, 153(1):29–46,2007.
9. Nikhil Bachhav , Priya Malode , Nirmala Bhujbal ,Shital Jawale. Dynamic Ride-Sharing Application on Android Platform. In Proc. Int'l Conf. Very Large Data Bases (PVLDB '14), pages 2017–2028, 2014.
10. Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Ride Matching Using K-means Method Social Networks, 5(3):269–287, 1983.
11. P. M. d'Orey, R. Fernandes, and M. Ferreira. Empirical evaluation of a dynamic and distributed taxi-sharing system. In Proc. IEEE Int'l Conf.

Intelligent Transportation Systems (CITS'12), pages 140–146, 2012.
12. A. Guttman. R-trees: A dynamic index structure for spatial searching. In Proc. ACM Int'l Conf. Management of Data (SIGMOD '84), pages 47–57, 1984.
13. K. Tsubouchi, K. Hiekata, and H. Yamato. Scheduling algorithm for on-demand bus system. Information Technology: New Generations, 2009.
14. B. Mcclosky and I. V. Hicks. Combinatorial algorithms for themaximum k-plex problem. Journal of Combinatorial Optimization, 23(1):29–49, 2012.

BIOGRAPHIES

Silpa K M, She is a post graduation student in College of Engineering Kidangoor. Specialization in Computer and Information Science. She received the graduation in Computer Science and Engineering from Govt. Engineering College, Palakkad. Her areas of interest are Big Data and Machine Learning.

Mary Priyanka K S, is an Assistant Professor in College of Engineering Kidangoor. Her areas of interest are Databases and Big Data.