GO!Caronas: fostering ridesharing with online social network, candidates clustering and ride matching

Michael Cruz
PROCC
Universidade Federal de Sergipe
São Cristóvão, Sergipe, Brazil
michaeloc9@gmail.com

Hendrik Macedo
DCOMP
Universidade Federal de Sergipe
São Cristóvão, Sergipe, Brazil
hendrik@dcomp.ufs.br

Erick Mendonça
DCOMP
Universidade Federal de Sergipe
São Cristóvão, Sergipe, Brazil
erick@erickmendonca.com.br

Adolfo Guimarães
Departamento de Ciência da Computação
Universidade Tiradentes
Aracaju, Sergipe, Brazil
adolfoguimaraes@gmail.com

ABSTRACT

The larger amount of motor vehicles produces many problems to cities around the world as poor urban mobility, environment concerns and people's health. The encouragement to use alternative means of transport, such as bicycles or subways is one of the steps to solve the problem. Another way is to encouraging the optimal usage of private automobiles. In Brazil, for instance, several studies show an underused space in cars and great potential for the use of carpooling and car sharing. Unfortunately, there is not such a unified technological mean to properly support these alternatives. In this paper we extend a carpooling social network with ridematching functionality in order to enable real-time ridematching. Furthermore, we implemented a functionality of group formation to encourage people to offer or get regular rides. We show that there is a favorable context to share rides between users. We describe the architecture of the system and implementation details. We present results about system's profiling and present a comparison of results related with acceptability of getting or offering ride between the years 2013 and 2015.

Categories and Subject Descriptors

H.4.0 [**Information System**]: Information System Applications - General

General Terms

Design System

EATIS'16, April 27–29, 2016 Cartagena de Indias, Bolivar, Colombia.

Keywords

Urban mobility, Carpooling, Real-time ridesharing, Ridesharing, ridematching, groups

1. INTRODUCTION

Traffic congestion is a serious concern in metropolitan areas [18]; health issues, environmental damages and economic losses are some of the known consequences [25], [8], [21], [16]. In most cities, road network could not stand the fast growth in the number of cars. According to the Brazilian national department of traffic, DENATRAN, the number of vehicles have increased more than 100% in the last 10 years [9]. In Beijing, China, where traffic is considered to be some of the world's worst, government has adopted the policy of restricting traffic for private cars. Even with this policy, though, traffic condition in peak hours is critical [17]. The city of São Paulo has adopted similar policy [5], but even so, it had some of the worst episodic of vehicle congestion in 2013 [12]. Since the U.S.A. suffered a loss of almost 78 billion in 2007 due to congestion issues [10], a lot of measures have been adopted to reduce the problem such as: building new roads/avenues, improve traffic light synchronization [17], encouraging the use of bicycles as daily transportation, improvements on public transportation and so on.

The congestion in large cities is a worrying reality [25]. The damage caused by traffic congestion reflects on various aspects of society, for instance, there is a loss in economic since there is no production during a congestion. Sectors such as delivery of goods suffer with constant delays. The cost of these problems are passed on to the population with the frequent increase in the shipping costs [25].

A common practice which is closely related to cultural habits in some nations and which can contribute to soften the problem is carpooling or ridesharing. Carpooling consists in sharing private vehicle space among people with similar destinations or daily trajectories [15], [22]. Sharing cars' empty seats is indeed such a kind of optimization procedure if we consider, for instance, the low occupancy rate per vehicles in traffic [17]. The mean occupancy of people per vehicle in U.S.A. transit in 2001 was 1.6.

More recently, in 2011, a research conducted by Michigan University has shown a occupancy rate of 1.5. Such occupancy rate is easily decreased to 1.4 when the trajectory is limited to “house-work” or “work-house”. According to [13], the average of rate occupancy, in England, between car and van is almost 1.5 between 2002 and 2014. In other words, there are many vehicles with just the driver inside [14]. From that, it is possible to conclude that the use of empty seats in vehicles might be an effective way to increase occupancy rate and as a result to soften traffic congestion.

There is a growing concern with the creation of new tools that promote the efficient usage of transport in order to avoid traffic jam and, as a consequence, provide a better environmental quality. There exists some software initiatives to facilitate the ridesharing’s practice such as UniCaronas [26], Caronas Brasil [3], BlaBlaCar [23], GO! [19], Tripda [27], RideWith [4], Carma [24], Carticipate [11], Uber [20], Lyft [28] are some examples.

Some services provided by these software require that interested users perform a search for people who offers a ride with the same or similar trajectories. In addition to the inherent difficulty in finding a corresponding ride, the driver and passengers are often unfamiliar. So, the so-called ridematching procedure has been proposed to deal with these issues and suggest the carpooling formation instantaneously [19]. It promises easing the matching process among candidates by properly assigning users who wish to get a ride to users that offer.

This paper proposes extending an on-line social network GO![22] to support the use of ridematching proposed by [7].This extension has been called GO!Caronas. The ridematching method used is a innovative approach to generate trajectory clusters based on POIs (Points of Interests) around the trajectory. Furthermore, GO! is integrated with a new functionality which permits that users can form groups of users with similar trajectories.

The rest of this paper is organized as follows. The Section 2 reviews systems with similar features. Section 3 presents the architecture and the main functionalities of the social network GO!Caronas. In section 4, we describe the ridematching approach. The Section 5 shows the results of the prospecting demand study and profile analysis of ridematching approach. Finally,in Section 6 we present some concluding remarks and future directions.

2. RELATED WORK

Carpooling is a specialization of ridesharing. Ridesharing is defined as grouping of travellers into common trip by car or van [6]. There exist some specialization of ridesharing, for instance, casual carpooling, real-time ridesharing, social networks and so on. Casual carpooling normally is formed during morning commute hours at park-and-ride facilities or public transit centers and takes advantage of existing HOV(High-Occupancy Vehicle) lanes to get to a common employment centre. Real-time ridesharing is more flexible than casual carpooling, because it uses mobile applications and automated ridematching software to organize ride in real time. This enables participants to be organized either minutes before the trip takes place or while the trip is occurring. The last kind of ridesharing uses social networks to find or match potential riders among friends. Social network platform are used with expectation that users can build trust and safety estimation [6].

Applications like Uber [20], lyft[28], Carticipate[11], Tripda [27], BlaBlacar[23] and Carma [24] are considered real-time ridesharing, because they do not require that participants know each other, in other words, require little relationship between participants and have some kind of ridematching method to find similar rides. Another characteristic is that most of these applications work like taxi and the idea the common trip is not so valid. Furthermore, some applications work like a kind of sophisticate taxi what is very different of ridesharing definitions.

The system CaronasBrasil [3] is one of the first systems for carpooling in Brazil. This application distinguishes of capooling definition gave by [6], because it is more flexible, in other words, users can get or offer ride in any time of the day. The Figure 1 shows a kind of classification among applications cited.

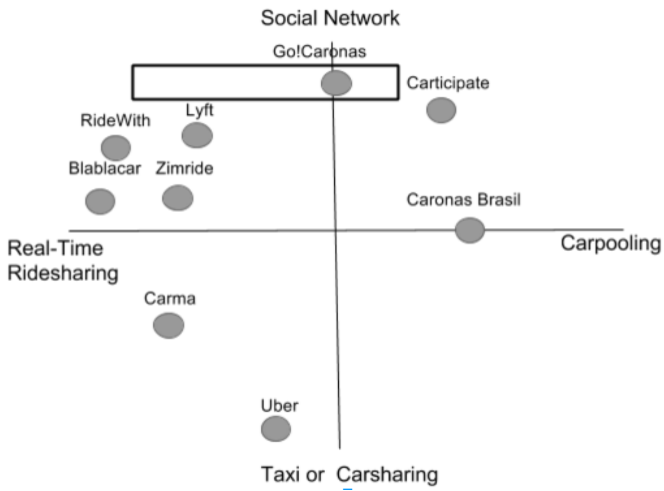

Figure 1: The comparison among applications.

The Figure 1 shows that great part of applications are classified as a real-time ridesharing and they have some of characteristics of social network, since those applications don’t build, but use social networks like Facebook[1], Goolge+ [2] etc. GO!Caronas application tries to keep three characteristics that have been judged important according to [6].

Another characteristics such as long distance (LD), short distance (SD), ridematching and profile matching can be other alternative to classify ridesharing applications. Long and short distance are related with length of trajectory that users normally share in some applications. Since ridematching and profile matching are characteristics related to the possibility that applications make matching automatically considering user’s trajectory, user’s profile or both. The Table 1 shows the classification of systems according to characteristics described.

GO!Caronas proposes to add ridematching approach to support real-time ridesharing, but the idea the common trip is maintained. Another functionality is to permit that people may create group with regular schedule in order to preserve the characteristic of carpooling.

[1]www.facebook.com
[2]www.plus.google.com

Table 1: Ridesharing applications

Applications	LD	SD	RMatching	PMatching
Carma	Yes	Yes	Yes	No
Tripda	Yes	No	Yes	No
Uber	Yes	Yes	Yes	No
BlaBlaCar	Yes	No	Yes	No
Lyft	Yes	Yes	No	No
CaronasBrasil	Yes	No	No	No
Carticipate	Yes	Yes	No	No
GoCaronas!	Yes	Yes	Yes	No
RideWith	Yes	Yes	Yes	No

Table 2: GO!Caronas functionalities

Functionalities	Old	New
User Registration	Yes	Yes
Rides Registration	Yes	Yes
Rides from friends	Yes	Yes
Rides from user	Yes	Yes
Request of rides	Yes	Yes
Confirmation and request messages	Yes	Yes
Public Profile	Yes	Yes
Reputation	Yes	Yes
Ridematching	No	Yes
Groups	No	Yes

3. GO!CARONAS

The system GO!Caronas extends the system GO! [22] which is a social network that allows the sharing of rides among users. The GO!Caronas adds ridematching algorithm in order to turn the social network in real-time ridesharing. Now, it is possible organize ridesharing in real time, just minutes before the trip takes place. Besides ridematching, the functionality of group formation was added to permit users build groups. Groups is a interesting functionality because permit that users invite users that work or study in the same place etc.

3.1 Architecture

The architectural pattern used in the project is the MVC (Model-View-Controller). The MVC divide the project into three layers: (1) model, (2) view and (3) controller [9].

The model layer is responsible for providing all communication with the database (logic and business rule). The view layer presents information to the user. This layer is not concerned with any kind of data treatment, but only to present information and receive user input. The control layer performs some treatments on the data received by the view layer and specifies the sequence of views. It is also responsible for handling the communication between the model and view layers.

The Figure 2 shows the main components of the architecture. We can observe the three MVC components responsible for the requests related to the systems model, vision and control.

The system implements two types of controllers: (1) Web Controller, which is implemented with the angularjs [3], (2) Mobile Controller, both controllers send or receive data in JSON format. These formats allow for greater data security. This controller uses the REST (Representational State Transfer) architecture [1].

3.2 System Description

Table 2 lists the old and new functionalities identified for a ridesharing system. These features have been identified through of analysis done in the major related systems. The Table 2 shows those that are available in the Web client and mobile client. An explanation of each functionality is presented below.

3.2.1 Ridematching

The driver must registrate a ride or a passenger must registrate some information, for example, departure and destination point, time and date. The ridematching algorithm will be capable of finding user with common interests, in other words, the system will find drivers and passengers with similar trajectory and schedule.

3.2.2 Groups

The users can create groups of people through ridematching algorithm or they can create manually. To create group through the algorithm, the users must registrate a ride and choose that the system find similar users automatically, so, the system will call the ridematching algorithm which will generate cluster based on the history of request or through the present requests. The ridematching algorithm may find users and it will show a list of result with users who can form groups. The another way to create a group is looking by users through the search.

3.3 Functional requirements

We illustrate two of the functional requirements of the system by means of activity diagrams [10]. The diagram of Figure 3 shows how to get rides suggestions through the ridematching, as explained below:

1. The user must register or ask rides.
2. The system will call ridematching algorithm that will look for similar user's trajectory.
3. After ridematching, a list of users with similar trajecotry will appear if algorithm found any results. On other hand, an alert message will appear to inform unsuccessful operation, in other words, ridematching didn't find users with similar trajectory and a list of random rides will appear.

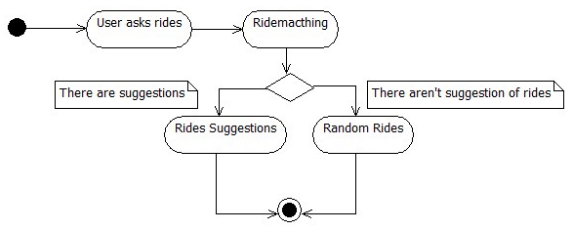

Figure 3: Activity diagram of ridematching operation.

[3]https://angularjs.org/

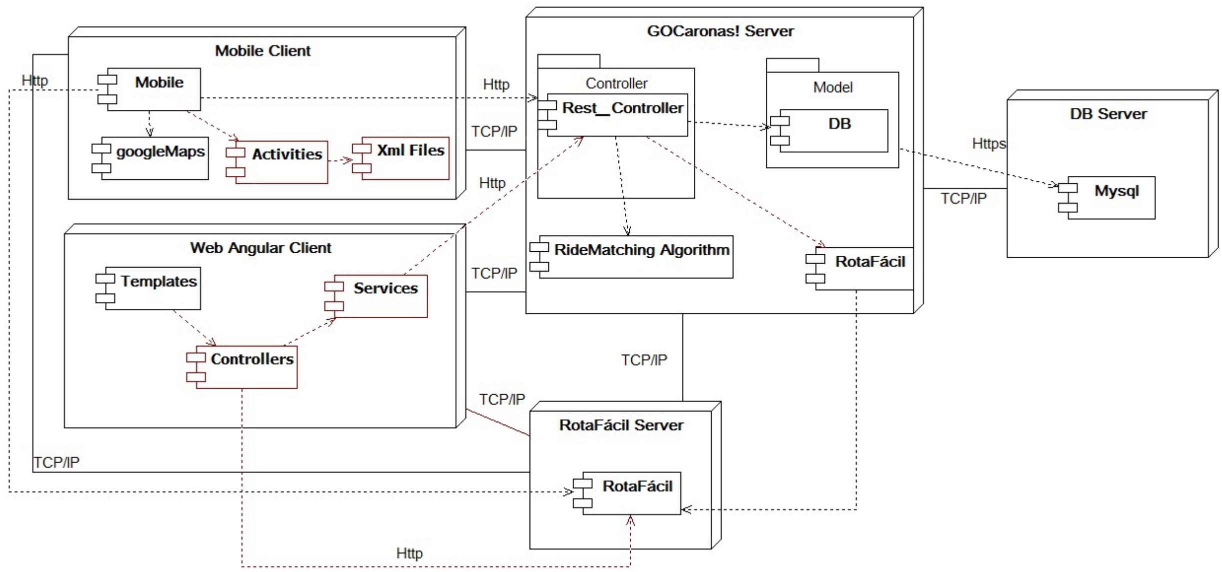

Figure 2: Architecture details GO!Caronas.

Grouping of users may be created through ridemacthing algorithm or manually. The diagram of Figure 4 shows how create groups:

1. User must register a schedule with destination and departure points.

2. User can choose how find others users, for example, automatically or not. First option enable the application to search for users as we have already said above. The another option, the user will search by friends that ask or offer rides with similar trajectory.

3. After find similar users, the application will present a list of them.

4. The user can invite similar users to form a group.

4. METHOD

Ridematching is a method to match similar rides. Similar rides, normally, is considered when users have similar trajectories and users have similar departure time. The ridematching approach used in GO!Caronas is defined by [7]. The Figure 5 shows how the approach works.

The approach is divided into three steps: (i) trajectory discretization, (ii) temporal filter and (iii) clustering. Trajectory discretization is used to reduce a quantity of trajectory points. Considering that database C is set of user's trajectory.

Trajectory is represented by points and each points is formed by triple $p_i = (lat, lng.timestamp)$. The discretization consists in computing a subset of points that is representative of the trajectory. The representative points are called POIs. The result of discretizantion enables the creation of the dataset $C^{`}$ which has only discrete trajectories according to the Figure 5.

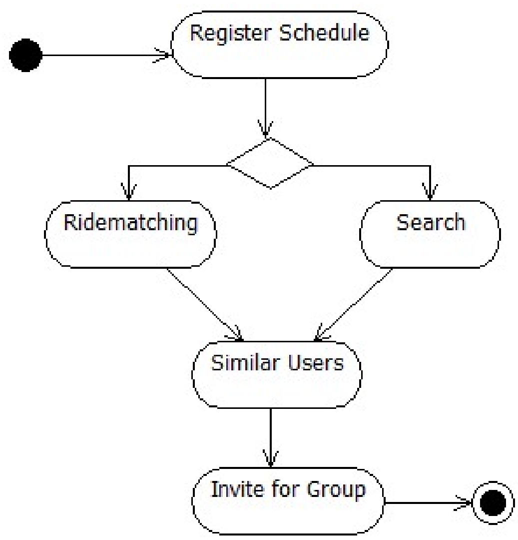

Figure 4: Activity diagram of grouping.

A temporal filter is established so that only the set of users that have trajectories with similar departure and destination times are eligible for the processing pipeline to avoid processing waste.

Clustering is used to grouping similar users's trajectory. According to work [7], cluster algorithm Optics [2] was adapted to group users with similar trajectories. This algorithm uses the similarity function defined in [7]. The function takes into account departure and destination points of passengers how

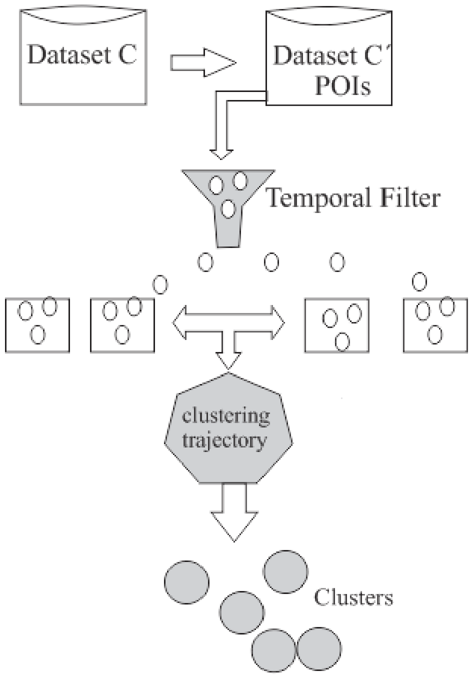

Figure 5: The method

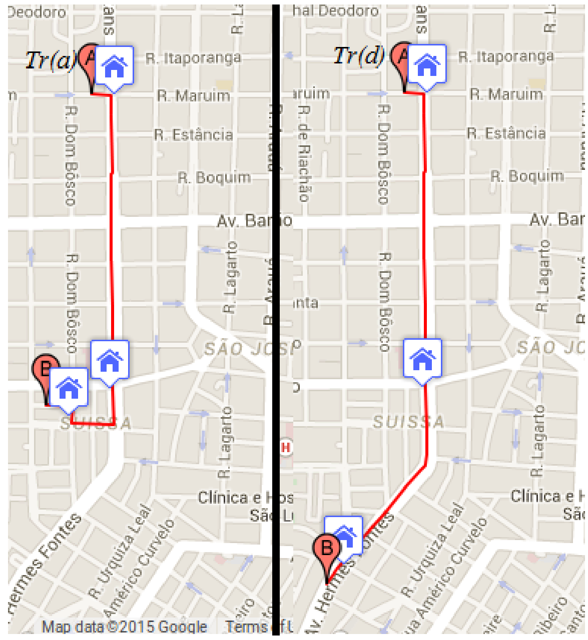

Figure 7: Real example of ridematching algorithm.

is shown on Figure 6.

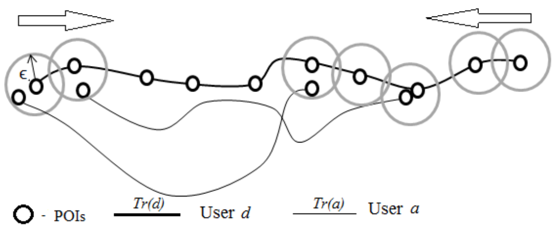

Figure 6: Approach used to define the similarity between two trajectories.

According to Figure 6, $Tr(d)$ belongs to driver user and $Tr(a)$ belongs to passenger. The main idea is to compare destination point of the passenger with all points of driver's trajectory and in the same way compare destination point. Those points are compared with all trajectory's points of driver in order to verify if passenger's trajectory and driver's trajectory are similar. The Figure 7 shows a real result which was reached by approach presented. The Figure 7 presents two similar trajectories that belong to the same cluster and represents two distinct users.

5. RESULTS

A Beta version of the GO!Caronas is available [4]. Figures 8 and 9 show some screenshots of the system. These pictures show just the new functionalities as a group of rides and rides which are automatically found by ridematching approach. The Figure 8 shows the interface of application to register a ride by group.

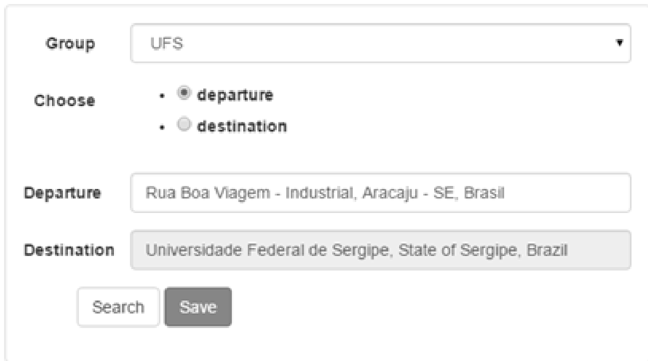

Figure 8: Screenshot of GO!Caronas - Register a ride in a group of ride.

The Figure 9 shows application's result related to a request done by user who has similar trajectory with another user.

We have reevaluated the system related with demand for carpooling service and evaluated the system through of the profiling approach. In the first case, we seek for the needs of potential users in regards to the services the system should provide. This evaluation compares results obtained, in 2013, with actual results.

In the second case, the system of ridematching has undergone automated profiling analysis to verify which function, classes or library has been a bottleneck or the frequency and duration of functions call.

[4] http://kb.erickmendonca.com.br:8000/go_caronas/login/

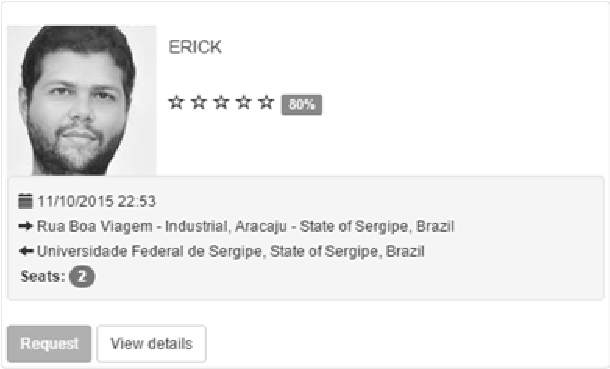

Figure 9: Screenshot of GO!Caronas - Example of result of ridematching.

5.1 Prospecting the need for ridesharing

We applied a questionnaire in order to find out what potential users think about and hope to get with a ridesharing on-line service and compared the current results with results obtained in [22]. More than 300 volunteers of Brazil were asked to respond the questions. The questionnaire was done and distributed through the Google Forms[5]. The Figure 10 presents the age of people who answered the questionnaire.

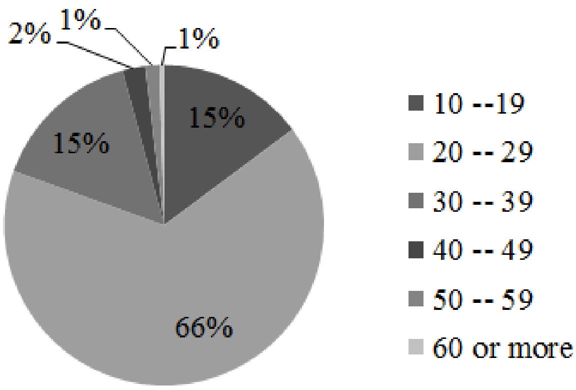

Figure 10: The range age of people.

Three first questions allows us to observe important aspects. The first question refers to the usual number of occupants in the car the user uses to daily work. The result is shown in the graphics of Figure 11. We can observe that the result is almost equal, in other words, great part of respondents use the car alone or with no more than one passenger. Both graphics of Figure 11 show that there are not big changes in the results.

The sencond question, we analyze the importance of cultural concerns to get and provide rides Figure 12.

Finally, in the third question, we analyze if the group functionality can encourage people use the system according to Figure 13.

[4]https://docs.google.com/forms/d/1XHGcHbfL7COvGmkcEurybc7rOwe_1ccebMpRrIRt7uk/viewform

[5]https://docs.google.com/forms

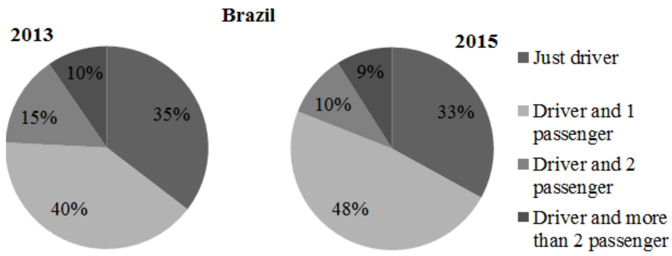

Figure 11: Number of occupants in cars

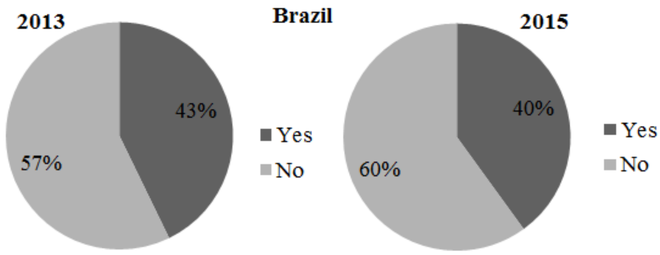

Figure 12: Cultural to deliver or get rides

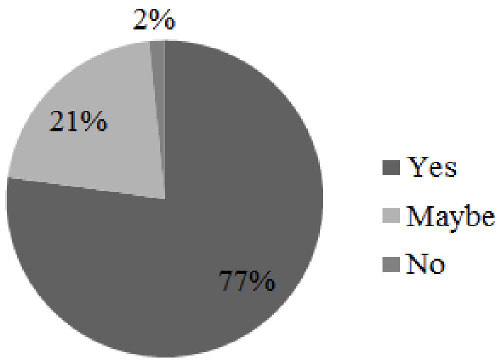

Figure 13: Form group may encourage user to get or offer ride.

5.2 Software profiling

The library cProfile [6] has been used to perform the profiling analysis. The analysis is done specifically in ridematching approach, because ridematching requires a lot of computation to find clusters with users with similar trajectories.

The analysis cover a set of features such as: a number of call of functions *NumofCall*, the total time spent by functions or operations, the cumulative time spent by the functions *CTime* etc. Besides, we checked the average of time that the method spends to run 500 trajectory of users was 25.019 seconds. The Table 3 shows three functions which are very expensive. Those functions are going to be improved.

[6]https://docs.python.org/2/library/profile.html

Table 3: Profiling ridematching analysis.

Function	Num of Call	Total Time	Cum Time
math.cos	13317696	1.846	1.846
distance	1902528	16.182	22.297
neighbors	408	0.150	24.223

The Table 3 shows that function *math.cos* has been called a lot of times, but the time expensive spent by the function is less than 10% of *neighbors* function. The neighbors function is used by Optics algorithm and has fundamental importance.

6. CONCLUSIONS

This work shows the extension of the system GO!, an online social network and tool for ridesharing. The main goal of the extension is to enable that the system find ride in real time. GO!Caronas has been built with purpose to increase the possibility of people find rides. As a result, the user may not only sharing rides manually but also is able to receive a list of users which have similar trips. The second extension supports the functionality of group formation that enable users form groups with fixed trips. The rationale is that the user would have a greater incentive to offer rides and as a consequence the society awareness of more rational usage of private transport.

The questionnaire has shown that personal automotive continue underused and people need to stimulate the habit of getting and providing rides. We can also conclude that people tend to use systems that may encourage such good habit although security concerns are considerable.

GO!Caronas is the second step towards building a platform that encourage ridesharing. As future work, we intend to provide integration with a new approach of ridematching which take into account trajectory and users profile to find similar rides. Finally, we intend to work on an improvement to enable personalized recommendation of rides and friends.

Acknowledgments

The authors thank Fundação de Apoio à Pesquisa e Inovação Tecnológica do Estado de Sergipe (FAPITEC-SE) for granting a scholarship to Michael Oliveira, the CNPq for granting a productivity scholarship to Hendrik Macedo [DT-II, Processo 310446/2014-7] and the Universidade Federal de Sergipe for the financial support [Edital POSGRAP/ COPES/ UFS No 03/2014 14/2012 (HERMES),Processo 008325/14-72].

7. REFERENCES

[1] S. Allamaraju. *Restful web services cookbook: solutions for improving scalability and simplicity.* " O'Reilly Media, Inc.", 2010.

[2] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod Record*, volume 28, pages 49–60. ACM, 1999.

[3] E. R. Azzam and F. D. D. Bellis. Carona Brasil. http://www.caronabrasil.com.br/, 2008. [Accessed: 06-September-2013].

[4] N. Bardin, E. Shabtai, A. Shinar, F. Shmuelevitz, Y. Elish, D.-A. Eisnor, and S. Keret. Ridewith. https://play.google.com/store/apps/details?id=com.ridewith, 2015. [Accessed: 30-July-2015].

[5] CET. CET - Rodízio Municipal. http://www.cetsp.com.br/consultas/rodizio-municipal/como-funciona.aspx, 2013. [Accessed: 14-September-2013].

[6] N. D. Chan and S. A. Shaheen. Ridesharing in north america: Past, present, and future. *Transport Reviews*, 32(1):93–112, 2012.

[7] M. Cruz, H. Macedo, and A. Guimarães. Grouping similar trajectories for carpooling purposes. In *Brazilian Conference on Intelligent Systems*. ACM, 2015.

[8] J. Currie. TRAFFIC CONGESTION AND INFANT HEALTH :. 2010.

[9] DENATRAN. Denatran - departamento nacional de trânsito. frota de veículos. http://www.denatran.gov.br/frota2013.html, July 2013. [Accessed: 10-January-2014].

[10] H.-E. Eriksson and M. Penker. Business modeling with uml. *Business Patterns at Work, John Wiley & Sons, New York, USA*, 2000.

[11] S. Frost. Carticipate. http://www.carticipate.com/, 2015. [Accessed: 30-July-2015].

[12] S. P. G1. São Paulo tem congestionamento recorde, com 300 km de filas. http://glo.bo/1bWMGE7, July 2013. [Accessed: 14-September-2013].

[13] S. P. G1. Car and van average occupancy rate in England between 2002 and 2014. http://www.statista.com/statistics/314719/average-car-and-van-occupancy-in-england/, 2014. [Accessed: 30-April-2015].

[14] K. Ghoseiri, A. Haghani, M. Hamedi, and M. Center. *Real-time rideshare matching problem.* 2011.

[15] R. Gowri. Car Pooling and Car Sharing: Simple Solution to Solve Complex Issues. http://www.frost.com/prod/servlet/market-insight-print.pag?docid=145008006, Aug. 2008. [Accessed: 14-September-2013].

[16] J. E. Hart, F. Laden, R. C. Puett, K. H. Costenbader, and E. W. Karlson. Exposure to traffic pollution and increased risk of rheumatoid arthritis. *Environmental health perspectives*, 117(7):1065–9, July 2009.

[17] W. He, K. Hwang, and D. Li. Intelligent Carpool Routing for Urban Ridesharing by Mining GPS Trajectories. 15(5):2286–2296, 2014.

[18] W. He, D. Li, T. Zhang, L. An, M. Guo, and G. Chen. Mining Regular Routes from GPS Data for Ridesharing Recommendations. pages 2–9, 2012.

[19] M. O. d. C. M. L. S. M. A. G. Hendrik, Texeira Macedo. Go!, 08 2014.

[20] T. Kalanick and G. Camp. Uber. https://www.uber.com/, 2015. [Accessed: 30-July-2015].

[21] J. I. Levy, J. J. Buonocore, and K. von Stackelberg. Evaluation of the public health impacts of traffic congestion: a health risk assessment. *Environmental health : a global access science source*, 9(1):65, Jan. 2010.

[22] M. L. Matos, M. Cruz, A. Guimarães, and H. Macedo. A social network for carpooling. In *Proceedings of the 7th Euro American Conference on Telematics and Information Systems*, page 10. ACM, 2014.

[23] F. Mazzella. Blablacar. http://www.blablacar.com, 2004. [Accessed: 06-September-2013].

[24] S. O'Sullivan. Carma. https://carmacarpool.com, 2015. [Accessed: 30-July-2015].

[25] P. T. V. RESENDE and P. R. SOUSA. Mobilidade urbana nas grandes cidades brasileiras: um estudo sobre os impactos do congestionamento. *SIMPOI–SIMPÓSIO DE ADMINISTRAçÃO DA PRODUçÃO, LOGÍSTICA E OPERAçÕES INTERNACIONAIS, FGV*, 2009.

[26] G. Souza and M. Marosti. UniCaronas. http://www.unicaronas.com.br/, 2009. [Accessed: 06-September-2013].

[27] A. Vaxman, P. Meduna, J. McFarlane, and E. Prota. tripda.com.br. http://www.tripda.com.br/, 2014. [Accessed: 12-November-2014].

[28] L. G. . J. Zimmer. Lyft. https://www.lyft.com/, 2015. [Accessed: 30-July-2015].