Grouping similar trajectories for carpooling purposes

Michael O Cruz
Computer Departament
Federal University of Segipe
São Cristóvão, Brazil

Hendrik Macedo
Computer Departament
Federal University of Segipe
São Cristóvão, Brazil

Adolfo Guimarães
Computer Departament
Federal University of Segipe
São Cristóvão, Brazil

***Abstract*—Vehicle congestion is a serious concern in metropolitan areas. Some policies have been adopted in order to soften the problem: construction of alternative routes, encouragement for the use of bicycles, improvement on public transportation, among others. A practice that might help is *carpooling*. Carpooling consists in sharing private vehicle space among people with similar trajectories. Although there exist some software initiatives to facilitate the carpooling practice, none of them actually provides some key facilities such as searching for people with similar trajectories. The way in which such a trajectory is represented is also central. In the specific context of carpooling, the use of Points of Interest (POI) as a method for trajectory discretization is rather relevant. In this paper, we consider that and other assumptions to propose an innovative approach to generate trajectory clusters for carpooling purposes, based on Optics algorithm. We also propose a new similarity measure for trajectories. Two experiments have been performed in order to prove the feasibility of the approach. Furthermore, we compare our approach with K-means and Optics. Results have showed that the proposed approach has results similar for *Davies-Boulding index* (DBI).**

Keywords*—*Carpooling, trajectory similarity, trajectory clustering.

I. Introduction

Vehicle congestion is a serious concern in metropolitan areas [1]; health issues, environmental damages and economic losses are some of the known consequences [2], [3], [4], [5]. 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 [6]. 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 [7]. The city of São Paulo has adopted similar policy [8], but even so, it had some of the worst episodic of vehicle congestion in 2013 [9]. Since the U.S.A. suffered a loss of almost $78 billion in 2007 due to congestion issues [10], a lot of measures has been adopted to reduce the problem such as: building new roads/avenues, improve traffic light synchronization [7], encouraging the use of bicycles as daily transportation, improvements on public transportation and so on.

A common practice which is closely related to cultural habits in some nations and which can contribute to soften the problem is *carpooling*. Carpooling consists in sharing private vehicle space among people with similar destinations or daily trajectories [11], [12]. 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 [7]. 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" trajectories. In other words, there are plenty of vehicles with just the driver inside [13]. 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 exists some software initiatives to facilitate the carpooling's practice; UniCaronas [14], Caronas Brasil [15], BlaBlaCar [16], carpooling.co.uk [17], Go! [12] and Tripda [18] are some examples. Some services provided by these softwares 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. To date, few carpooling software have had commercial success [13]. Some of the essential features are the easy of use, safety, flexibility and efficiency. There are other important factors that may discourage carpooling: smoking, features of the vehicle itself, social and demographic profile of the driver [19] and gender [20]. The so-called *ridematching* procedure has been proposed to deal with the 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 offers.

Currently, the proliferation and the ease to use smartphones apps, the *Global Position System* (GPS) and APIs like *GoogleMaps* [1] enable people to track their own trajectories and share them widely: *Bikely* [2], *GPS-Way-Points* [3], *Share-My-Route* [4], *Microsoft Geolife* [5], *Facebook* [6] [21]. These shared data can be used to the development of a lot of interesting features such as: mining frequent trajectories [22], finding similar trajectories [23], mining points of interests (POIs), find out sub-trajectories, and so on.

[7] and [24] have tried different approaches to provide ridematching among users based on trajectory mining. Indeed, most research focuses on the improvement of the trajectory mining process but few propose an effective approach to discretize the GPS-based trajectory.

[1]https://www.google.com.br/maps
[2]http://www.bikely.com/
[3]http://www.gps-waypoints.net/
[4]http://www.sharemyroutes.com/
[5]http://research.microsoft.com/en-us/projects/geolife/
[6]http://www.facebook.com/

978-1-5090-0016-6/15 $31.00 © 2015 IEEE
DOI 10.1109/BRACIS.2015.36

We argue that in the specific context of carpooling, using POIs as a method for trajectory discretization is rather relevant since they can decrease redundant points and the amount of processed data. In particular, it is worth noting that people use POIs to describe trajectories in spite of the use of GPS points. We also argue that the carpooling context softens the constraints for ridematching since the departure and destination points of users that receive the ride do not need to be exactly the same as the users that offer the rides, but shall be comprised between or near them.

In this paper, we propose an innovative approach to generate trajectory clusters in the context of carpooling based on POIs around the trajectory. We adapted the Optics [25] algorithm with an innovative approach of similarity measure to group trajectories. Departure time is obviously an important feature and should be worked as a filter.

The rest of this paper is organized as follows. Section 1 reviews clustering techniques for trajectories. In section 2, we describe our approach to trajectory clustering. Experiments and results are presented in section 3. Finally, we conclude the work in section 4.

II. Trajectory Clustering

Clustering is the process of grouping a set of physical or abstract entities within similar categories [26]. According to [27] and [28], clustering can be used to provide:

- *data Reduction*, when the amount of available data is huge and, as a consequence, processing becomes very expensive; clustering can be used to generate representative data groups and further processing these data;
- *hypothesis generation* in regards to data nature;
- *hypothesis testing*, by raising a hypothesis and, thus, clustering the data; clusters can or cannot confirm the hypothesis raised;
- *prediction based on groups*: In this case, cluster analysis is used to available the data set and characterize every cluster based on the features of the patterns by which they are formed. The clusters results are used to characterize a new unknown pattern according with the similarity between them.

Clustering algorithms can be classified in: partitioning, hierarchical and density-based. The first group builds the partition of a data set taking into account the parameter k which defines the number of clusters that can be formed. Two known algorithms of this group are *k-means* and *k-medoid* [28]. Hierarchical algorithms decompose a data set hierarchically and can be represented by a dendogram, which is a tree that recursively divides the data set into smaller subsets towards unit sets. There are two distinct approaches for hierarchical algorithms: agglomerative and divisive [28]. Density-based algorithms discover clustering according to density of data. These algorithms view the clusters as regions in the n-dimensional space, in other words, they have ability to find clusters in multidimensional data. Besides that, density-based algorithms needs a minimal number of inputs parameters, as well as they can discover clusters of arbitrary shapes [28],[29]. Examples of density-based algorithms are DBSCAN, Optics, ADACLUS [30], [25], [31].

Some of these algorithms might be applied for clustering trajectories of any moving entity like cars or animals, for instance. Indeed, there are some research in trajectory clustering. [24] proposes a framework to assist in the discovery of sub-trajectories that are similar. The work also contributes by proposing a clustering algorithm called TRACULS. The algorithm performs two steps: (1) partitioning and (2) clustering. Partitioning divides a trajectory, which is defined as a sequence of points $RT = p_1, p_2, p_3, ..., p_{n-1}, ..., p_n$. In the clustering step, lines that are similar to representative trajectory are grouped into a cluster. Figure 1 summarizes framework's operation.

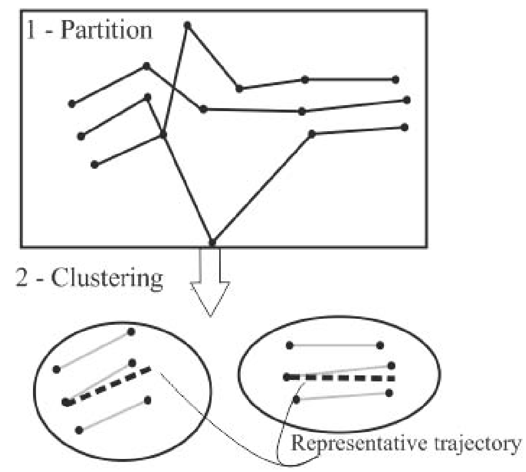

Fig. 1. An example of sub-trajectory clustering.

[32] has introduced the concept of *moving clusters*, which are set of entities that move close and do not lose their density throughout their lifetime. These clusterings appear during some time so that the number of objects, that are common, is not below a certain limit θ. The author says that $c_t c_{t+1}$ is a cluster if $\frac{|c_t \cap c_{t+1}|}{|c_t \cup c_{t+1}|} \geq \theta$ considering c_t a grouping time t as Figure 2.

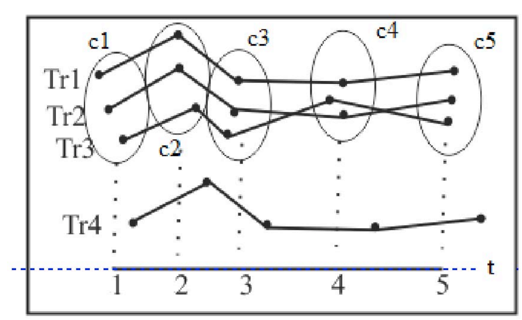

Fig. 2. An example of the movie cluster

[33] uses the Optics algorithm to group trajectories. According to [34], Optics is well suited to trajectory clustering

due to two main features: ability to build non-spherical clusters and sturdiness to noise.

III. Method

Our approach to Optics works as follows. Given a set of trajectories $C = \{Tr_1, Tr_2, ..., Tr_n\}$, it generates a set of clustering $A = \{C_1, C_2, ..., C_n\}$, where each C_i has at least one trajectory from a user that provides a ride. Trajectory is defined as $Tr_i = p_1, p_2, p_3, ..., p_n$, where each p_i is a triple (latitude, longitude, time).

Figure 3 depicts the whole method. Note that clustering process takes into account just special distances among trajectories; spatial distance and social distance is subject of a further paper.

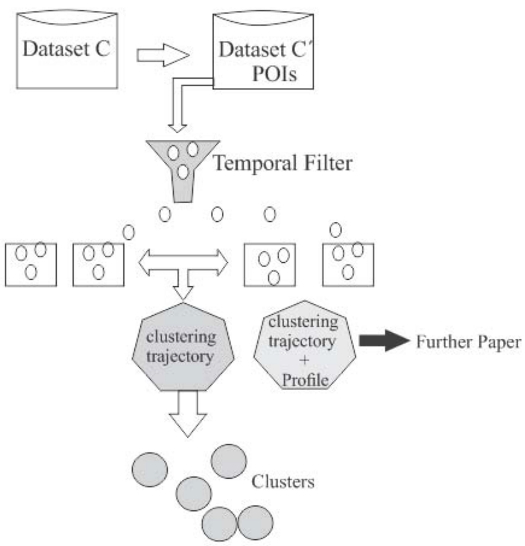

Fig. 3. Method

The method is divided into three steps: (i) trajectory discretization, (ii) temporal filter and (iii) clustering. Next subsections detail each of them.

A. Trajectory's discretization

Consider C a set of trajectories that have a large number of points. Many of such points are redundant due to the short time interval in which they were obtained. Trajectory's discretization consists in computing a subset of points that is representative enough of the trajectory. RotaFacil [35], [36] has been used for that. RotaFacil automatically generates natural language route descriptions between two given points using the set of POIs it finds trhoughout the trajectory. Given a radius threshold, it is also capable of identifying the POIs within that circumference (Figure 4). In our case, we have empirically defined the radius threshold as 200m (two hundred meters). The result of RotaFacil's processing is a subset C'.

B. Temporal filter

A temporal filter is established so that only the set of trajectories with similar departure and destination times are

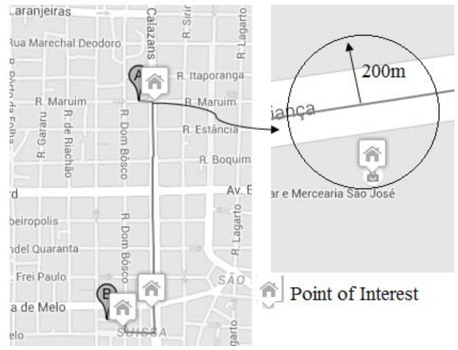

Fig. 4. POIs within a circumference for a given radius threshold.

eligible for the processing pipeline to avoid processing waste. Indeed, it is pointless clustering together Tr_u and Tr_v if departure and/or destination times of users u and v are rather distinct, even though $Tr_u \simeq Tr_v$. Considering t the time of such a ride offering, the width of the filter is the interval $[t - x, t + x]$ (Figure 5).

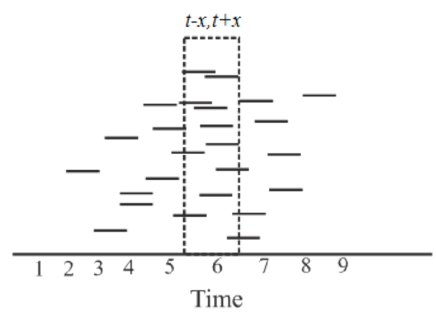

Fig. 5. Temporal filter to trajectory clustering.

C. Clustering

For Optics algorithm, *density* is defined by two parameters: (i) ε, the radius of the search and (ii) *MinPts*, the minimum number of neighbours that defines a cluster. In addition to these parameters, we had to compute the distance d between two points, defined by

$$d = \textit{Haversin}^{-1}(h) = 2R \arcsin \sqrt(h) \tag{1}$$

where h is the *Haversine*(d/R):

$$\textit{Haversine}(\frac{d}{R}) = \sin^2(\frac{\delta\phi}{2}) + \cos(\phi_1)\cos(\phi_2)\sin^2(\frac{\delta\gamma}{2}) \tag{2}$$

where R is the radius of the earth, d the distance between two points, $\delta\phi$ is the difference between the latitudes and $\delta\gamma$ is the difference between the longitudes.

Distance function is used to compute the similarity among trajectories. Similarity is defined considering that in such a ride, there is no motivation for user u, who offers the ride, deviates from his original trajectory to give a ride to a user v. The algorithm for similarity computation between two trajectories A and B is shown bellow.

Algorithm 1 Similarity algorithm

```
procedure SIMILARITY(Tr_u, Tr_v)        ▷ Tr_u and Tr_v are trajectories
    flagA ← FALSE
    flagB ← FALSE
    first ← getFirstPoint(Tr_v)
    last ← getLastPoint(Tr_v)
    for u_i ∈ Tr_u,i=1,..n do
        if dist(u_i, v_first) ≤ ε then
            flagA ← TRUE
        end if
    end for
    for u_i ∈ Tr_u,i=n,..1 do
        if dist(u_i, v_last) ≤ ε then
            flagB ← TRUE
        end if
    end for
    return flagA ∧ flagB
end procedure
```

Figure 6 summaries the algorithm behaviour. Firstly, it verifies if the trajectory Tr_v that belongs to a user v who wishes to get a ride is similar to trajectory Tr_u that regards the offering ride. Similarity takes into account only departure and destination points of the trajectory of the user who wish to receive a ride and tries to check if these points are within the radius of one of the points of the target trajectory.

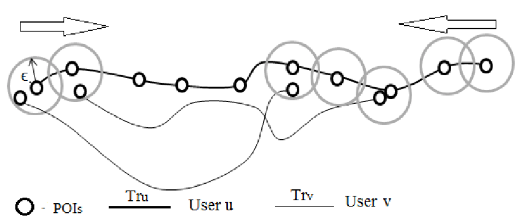

Fig. 6. Approach used to define the similarity between two trajectories.

Similarity algorithm defines the neighbourhood for each trajectory. It is also used in another procedure to save two pieces of information for every trajectory: (i) the *core-distance* and (ii) the *reachability-distance*. The *core-distance* is defined as the shortest distance between one Tr_i and its neighbours. The *reachability-distance* of a Tr_i to another Tr_j is the shortest distance such that Tr_i is *directly density-reachable* from Tr_j if it is a *core* object. A trajectory Tr_i is a *core* object if the number of its neighbours $Nb_i \geq MinPts$. Tr_i is *directly density-reachable* of Tr_j if $Tr_i \in Nb_j$ and $Nb_j \geq MinPts$.

The execution of the Optics does not produce a clustering explicitly, but instead creates an augmented ordering database representing its density-based clustering structure. The algorithm uses an infinite number of distance parameters $\varepsilon^{'}$ which are smaller than the generating distance ε. It is through the value *directly density-reachable* that clusters are extracted. The phase of extraction cluster is not part of Optics and, thus was necessary to define a new parameter which is a distance values $\varepsilon^{'}$, such that $\varepsilon^{'} \leq \varepsilon$ to check for clusters according to the $\varepsilon^{'}$ informed.

IV. Experiments

We have performed two experiments in order to prove the feasibility of the approach to trajectory clustering in the context of carpooling.

A. Datasets

We use two different datasets. First dataset consists of actual trajectories collected from users of the Go!Track[7] app [37]. GO!Track continuously collects GPS points of the trajectories people are taking while in theirs cars. This set currently contains around 40 trajectories with more than 5,317 points. Second dataset consists of 500 trajectories, artificially generated by RotaFacil.

B. Experimentation setup

Discretization applied to the first dataset has reduced the original 5,317 points to 394, considering the radius of 200 meters in RotaFacil.

In order to generate the artificial trajectories, $n = 23$ addresses were randomly chosen. Every address consisted of a departure point and a destination point. The number of different trajectories was then $n^2 - n$.

For both experiments, we have varied ε Optics parameter with 100m, 200m, 300m and $MinPts$ parameter with 2 and 3. We have assumed that 100 to 300 m are reasonable distance limits for a user who desires a ride to move towards the destination point of a offering ride. For the cluster extraction algorithm, the values were set to 50, 150 to parameter $\varepsilon^{'}$.

Experiments were executed with adapted Optics ($Optics^*$), which was modified to work with similarity function defined, original Optics and K-means. The K-means was applied with the same number of cluster discovered by Optics.

C. Evaluation metrics

Davies-Boundin index (DBI) [28] was used to evaluate the clustering task. Equation 3 defines the DB value:

$$DBI = \frac{1}{n}\sum_{i=1}^{n} max_{i \neq j}(\frac{\alpha_i + \alpha_j}{d(c_i, c_j)}) \quad (3)$$

where n is the number of clusters, c_i and c_j is the centroid of each cluster. The α_i and α_j are the similarity measures for clusters c_i and c_j.

The values generated by Equation 3 reflect how similar the elements of the same cluster are, as well as the dissimilarity among distinct clusters. Smaller DBI values are better.

[7]https://play.google.com/store/apps/details?id=com.go.router

D. Experimentation results

Table I shows the results of the first experiment. The number of clusters *NC* is zero when $MinPts$ is 3. According to the literature [25], the number of ε heavily influences the number of clusters, but $MinPts$ may provide some influence as well.

Zero values occurred in Table I, because the number of neighbors of the user trajectories was less than 3. This show that the real base have not many similar trajectories. Besides, Table I doesn't show K-means and Optics results, because they didn't discover clusters.

TABLE I. RESULTS FOR THE DATASET WITH ACTUAL TRAJECTORIES.

$Optics^*$	ε	$MinPts$	ε'	NC	DBI
	200	2	150	3	0,2885
	200	3	150	0	0
	300	2	250	3	0,2885
	300	3	250	0	0

Table II shows that the DBI values among three algorithms are similar. The DBI values demonstrates which clusters were very concise, in other words, the internal distance between user trajectories of the same group were too small and the distances among clusters were large. Furthermore, the number of clusters also increased when ε increased accordingly, as expected [25].

TABLE II. RESULTS WITH ARTIFICIALLY GENERATED DATASET

$Optics^*$	ε	$MinPts$	ε'	NC	DBI
	100	2	50	39	0,7868
	200	2	150	54	0,8860
	200	3	150	34	0.7235
	250	2	150	62	0,8968
	250	3	150	36	1,0164
	300	2	100	48	0,7946
Optics					
	100	2	50	18	0,0910
	200	2	150	47	0,2306
	200	3	150	20	0,3702
	250	2	150	47	0,2339
	250	3	150	20	0,3669
	300	2	100	29	0,1635
K-Means					
	0	0	0	18	0,7374
	0	0	0	47	0,4303
	0	0	0	20	0,6802
	0	0	0	47	0,4303
	0	0	0	20	0,5731
	0	0	0	29	0,5832

ε directly influences clusters' size according to Table II. Any ε that is "big" enough will produce good results. Unlikely, small ε will produce a lot of objects with *reachability-distance* value equal to *undefined*. In this work, neither method was used to deduce the perfect ε.

$MinPts$ values were also chosen randomly. Although the results of the Table I show that the number of clusters varies according to the values of $MinPts$, the work [25] affirms that high values for $MinPts$ can cause a *single-link effect* which is defined as a weak connection among distinct clusters.

TABLE III. KRUSKAL-WALLIS TEST

Method	N	Median Posts
$Optics^*$	11	6,73
Optics	11	24,18
K-Means	11	20,09
Total	33	
Sig.	,000	

Though Table II shows that there are differences among DBI values for three methods, Table III and Table IV show that Optic* performs as good as Optics and K-means taking into account a significance level set to $\alpha = 0,5$. Table III presents results according to Kruskall-Wallis which is non parametric test in order to compare two or three population to verify if there are differences among distributed function of each population. The second test, Median test, was used to reinforce the first test.

TABLE IV. MEDIAN TEST

	$Optics^*$	Optics	K-means
$> Median$	0	10	6
$\leq Median$	11	1	5
Sig.	,000		

V. CONCLUSION

Encouraging carpooling is an important effort towards the reduction of in-transit vehicles. Although there is some concerned research initiative and even some related software, they do not appropriately treat carpooling context specificities. In this paper, we have proposed a method to deal with some of these specificities such as temporal and distances concerns and Points of Interests (POIs) in a city as the users mainstream approach to describe their own trajectories. In particular, the method focuses on clustering similar users trajectories in order to discover potential carpooling opportunities.

Clustering results and corresponding *Davies-Bouldin Index* values obtained from a dataset of actual trajectories collected pervasively have shown the feasibility of the proposal. Moreover, significance analysis reinforce the feasibility presented by DBI metric. The results also show that POIs perform well to discretize trajectories in carpooling context.

We are currently working on enriching users profile with social and demographic information in addition to trajectories. We also intend to develop a carpooling recommender service so it could be integrated to some carpooling software, such as the GO! Caronas [12], [38].

ACKNOWLEDGMENT

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

REFERENCES

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

[2] 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.

[3] J. Currie, "TRAFFIC CONGESTION AND INFANT HEALTH :," 2010.

[4] 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*, vol. 9, no. 1, p. 65, Jan. 2010. [Online]. Available: http://www.pubmedcentral.nih.gov

[5] 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*, vol. 117, no. 7, pp. 1065–9, Jul. 2009. [Online]. Available: http://www.pubmedcentral.nih.gov/

[6] DENATRAN, "Denatran - departamento nacional de trânsito. frota de veículos," http://www.denatran.gov.br/frota2013.htm, Jul. 2013, [Accessed: 10-January-2014].

[7] W. He, K. Hwang, and D. Li, "Intelligent Carpool Routing for Urban Ridesharing by Mining GPS Trajectories," vol. 15, no. 5, pp. 2286–2296, 2014. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6812224

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

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

[10] D. L. Schrank and T. J. Lomax, *The 2007 urban mobility report*. Texas Transportation Institute, Texas A & M University, 2007.

[11] 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].

[12] 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*. ACM, 2014, p. 10.

[13] K. Ghoseiri, A. Haghani, M. Hamedi, and M. Center, *Real-time rideshare matching problem*, 2011. [Online]. Available: http://ntl.bts.gov/lib/44000/44900/44921/UMD-2009-05.pdf

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

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

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

[17] S. Weber, M. Reinicke, and M. Siedler, "Carpooling.co.uk," http://www.carpooling.co.uk/, 2000, [Accessed: 06-September-2013].

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

[19] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, Dec. 2012. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0377221712003864

[20] I. P. Levin, M. Mosell, C. Lamka, B. Savage, and M. Gray, "Measurement of psychological factors and their role in travel behavior," *Transportation Research Record*, no. 649, 1977.

[21] S. Shang, R. Ding, B. Yuan, K. Xie, K. Zheng, and P. Kalnis, "User oriented trajectory search for trip recommendation," in *Proceedings of the 15th International Conference on Extending Database Technology*. ACM, 2012, pp. 156–167.

[22] N. S. Savage, S. Nishimura, N. E. Chavez, and X. Yan, "Frequent trajectory mining on GPS data," *Proceedings of the 3rd International Workshop on Location and the Web - LocWeb '10*, pp. 1–4, 2010. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1899662.1899665

[23] N. Pelekis, I. Kopanakis, G. Marketos, I. Ntoutsi, G. Andrienko, and Y. Theodoridis, "Similarity Search in Trajectory Databases," *14th International Symposium on Temporal Representation and Reasoning (TIME'07)*, pp. 129–140, 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4438678

[24] J.-g. Lee and J. Han, "Trajectory Clustering : A Partition-and-Group Framework ," 2007.

[25] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod Record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.

[26] J. Han, M. Kamber, and J. Pei, *Data mining, southeast asia edition: Concepts and techniques*. Morgan kaufmann, 2006.

[27] G. H. Ball, "Classification analysis." 1970.

[28] S. Theodoridis, A. Pikrakis, K. Koutroumbas, and D. Cavouras, *Introduction to Pattern Recognition: A Matlab Approach: A Matlab Approach*. Academic Press, 2010.

[29] M. Parimala, D. Lopez, and N. Senthilkumar, "A survey on density based clustering algorithms for mining large spatial databases," *International Journal of Advanced Science and Technology*, vol. 31, no. 1, 2011.

[30] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[31] G. V. Nosovskiy, D. Liu, and O. Sourina, "Automatic clustering and boundary detection algorithm based on adaptive influence function," *Pattern Recognition*, vol. 41, no. 9, pp. 2757–2776, 2008.

[32] P. Kalnis, N. Mamoulis, and S. Bakiras, "On discovering moving clusters in spatio-temporal data," in *Advances in spatial and temporal databases*. Springer, 2005, pp. 364–381.

[33] S. Kisilevich, F. Mansmann, M. Nanni, and S. Rinzivillo, "Spatio-Temporal Clustering: a Survey," *Data Mining and Knowledge Discovery Handbook*, pp. 1–22, 2010. [Online]. Available: http://www.springerlink.com/content/m32658456701p668/

[34] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," *Journal of Intelligent Information Systems*, vol. 27, no. 3, pp. 267–289, 2006.

[35] R. Telles, A. P. Guimarães, and H. T. Macedo, "Automated Feeding of POI Base for The Generation of Route Descriptions," in *Euro-American Conference on Telematics and Information Systems, EATIS 2012*, 2012, pp. 253–259.

[36] R. Teles, B. Barroso, A. Guimaraes, and H. Macedo, "Automatic generation of human-like route descriptions: a corpus-driven approach," *Journal of Emerging Technologies in Web Intelligence*, vol. 5, no. 4, pp. 413–423, 2013.

[37] GOOGLE, "Google Play Store," https://play.google.com/store/apps/, 2013, [Accessed: 20-May-2015].

[38] M. O. d. C. M. L. S. M. A. G. Hendrik, Texeira Macedo., "Go!" Registro de Software BR 51 2014 000 963 7, 08 15, 2014.