Original Research Paper

Measuring the Relevance of Trajectory Matching and Profile Matching in the Context of Carpooling Computational Systems

1Michael Cruz 1Hendrik Macedo and 2Adolfo Guimarães

1Pós-graduação em Ciência da Computação, Universidade Federal de Sergipe, São Cristóvão/SE, Brazil
2Departamento de Computação, Universidade Tiradentes, Aracaju/SE, Brazil

Abstract: Carpooling consists of sharing individual vehicle space among people with comparable trajectories. Although there are some software initiatives to help carpooling practice, none of them really implements features similarly to searching for people with similar trajectories and profile. In this study, we propose an innovative approach to generate clusters of users that share similar trajectories and profile for carpooling purposes based on Optics, K-means algorithm and ensemble learning. First, we provide a proper definition of fundamental elements of the carpooling context in order to contribute to a standardization of the concerning nomenclatures. Next, we perform four different experiments for the purpose of showing the feasibility of the approach. We also contribute to the construction of a real dataset (donated to UCI), properly depicted, used in two of these experiments. Results with Davies-Boulding indicate that the generated clusters are feasible to the design of a carpooling recommendation system. Time performance evaluation of the approach has been also performed for both dynamic program analyses via software profiling method and time complexity analysis according to Big O notation.

Keywords: Carpooling, Trajectory Similarity, User Profile Similarity, Clustering

Introduction

Traffic jams are a serious concern in metropolitan areas (He et al., 2012). Economic losses, health issues and environmental damages are some of the known consequences (Resende and Sousa, 2009; Currie and Walker, 2010; Levy et al., 2010; Hart et al., 2009). According to the DENATRAN (Brazilian National Department of Traffic), the number of vehicles has increased more than 100% in the last 10 years (DENATRAN, 2013). An alternative to avoid the congestion is to adopt the policy of restricting traffic for private cars. For example, in Beijing, China, the government has adopted these solutions to solve the problem of the one worst traffic in the world. Although it is a solution, the traffic is still critical in peak hours. (He et al., 2014). Since the USA suffered a loss of almost $78 billion in 2007 due to traffic jam issues (Schrank and Lomax, 2007), a lot of measures has been adopted to reduce the problem such as: Improve traffic light synchronization (He et al., 2014), building new roads/avenues, encouraging the use of bicycles as daily transportation and improvements in public transportation.

Carpooling (share individual vehicle space among people with similar destinations) is a typical solution used by some nations to avoid the problems generated by the increase traffic condition. However, this solution is strongly related to some cultural aspects (Gowri, 2008; Matos et al., 2014). Sharing cars’ empty seats may be seen as an optimization method if we consider, for instance, the low occupancy rate per vehicles in traffic (He et al., 2014). In 2011, a research conducted by the Michigan University has shown an occupancy rate of 1.5 in the U.S.A. Such occupancy rate is easily decreased to 1.4 if we consider only “home->work” or “work->home” trajectories. In other words, there are plenty of vehicles with just the driver inside (Ghoseiri et al., 2011).

There are software initiatives to facilitate the carpooling’s practice. Caronas Brasil (Azzam and Bellis, 2008), Zumpy, PoolmyRide, BlaBlaCar (Mazzella, 2004), Go! (Matos et al., 2014), Carma (O’Sullivan, 2015), Carticipate (Frost, 2015), BeepMe, Lyft (Zimmer and Logan, 2012) and Bynd are some examples. However, to
date, few of them have had commercial success (Ghoseiri et al., 2011). Some services provided by that software require that interested users execute a search for people who offer a ride with the same or alike trajectories. In some cases, it’s not so simple to find this trajectory. Another problem is the fact that the driver and the passenger are unfamiliar. This kind of information contributes to the increasing of trust among users of the services (Furuhata et al., 2013). There are other factors that may discourage carpooling: The presence of a smoker, features of the vehicle itself, some aspects of the driver’s profile (Agatz et al., 2012) and gender (Levin et al., 1977). The ridematching procedure has been proposed to deal with these issues and suggest the carpooling formation instantaneously (Agatz et al., 2012). It promises to facilitate the matching process among candidates by correctly attributing users who want to get a ride to users that offers.

Currently, the propagation and the facility of use of smartphone apps, the Global Position System (GPS) and APIs like GoogleMaps allow people to track their own trajectories and share them broadly: GPS-Way-Points, Share-My-Route, Bikely, Facebook (Shang et al., 2012). These shared data can be used to the development of a lot of impressive characteristics such as: Mining frequent trajectories (Savage et al., 2010), finding similar trajectories (Pelekis et al., 2007), mining Points of Interests (POIs) (Telles et al., 2012), find out sub-trajectories and so on. He et al. (2014) and Lee et al. (2007) have tried different approaches of mining trajectory to provide ridematching among users. Surely, most research centers on the improvement of the trajectory mining process, but few propose an effective approach to define the suitable granularity level of GPS-based trajectory and none have properly formalized central elements and features of carpooling context. As a consequence, a range of terms with the same meaning is used in different works and may confuse the reader: Route (He et al., 2014) or trajectory (Lee et al., 2007), a driver (Furuhata et al., 2013), passenger or riders (Agatz et al., 2012), etc. Finally, in the context of carpooling, few academic types of research consider both similarity of profile and trajectory (Furuhata et al., 2013; Yan and Chen, 2011).

Cruz et al. (2015) propose a clustering approach to trajectories in the context of carpooling. This paper aims to extend the work of (Cruz et al., 2015) along three axes: (i) Propose (semi-formal) definition of the elements of the carpooling context, towards a standardized nomenclature, (ii) Add users’ profiles to the clustering approach and (iii) Provide a proper evaluation of the trajectories’ dataset (GO! Track) used to train the clustering model.

In section 2 we provide proper definitions to the elements of carpooling context. In Section 3, we describe our approach to generating user’s clusters with similar profiles and trajectories. In Section 4, we present the experiments and discuss the results. We conclude the work in section 5.

**Formalization**

**Definition 1:**

Trajectory is a sequence of multi-dimensional points. These points are discrete and finite and they are represented by \( Tr = \{p_1, p_2, p_3, \ldots, p_n\} \). Here, \( p \) is a 3-dimensional point: Latitude, longitude and time-stamp, \( p = \{lat, lng, t\} \).

**Definition 2:**

Driver is the user who shares a vehicle with the passenger and has similar trajectory with all passengers. \( Tr(d) \) is a trajectory that pertains to the driver. In this study, \( Tr(d) \sim U = \{T r(a_1), (a_2), \ldots, T r(a_n)\} \) means that the driver’s trajectory and passengers’ trajectory are similar. In such case, \( dist(Tr(d), Tr(a_i)) \leq r \), where \( dist() \) is some distance function and \( r \) is a limit constant.

**Definition 3:**

A vehicle is defined as any means by which someone may travel: A car, a motorcycle, etc. Here, a vehicle is represented by \( V \), where \( V(d) \) is a vehicle that belongs to the driver \( d \).

**Definition 4:**

Passenger is a user who shares a vehicle with a driver. \( Tr(a) \) is a trajectory that belongs to a passenger \( a \). In this study, \( Tr(a) \sim T r(d) \) means that the driver’s trajectory and passenger’s trajectory are similar.

**Definition 5:**

Ride is described as a form to share a private vehicle space among people with similar trajectory and interests. A ride is represented by \( R = (V(d), d, Tr(d), A) \), where \( V(d) \) is driver’s vehicle, \( d \) is a driver, \( Tr(d) \) is a driver’s trajectory and \( A \) is the set of passengers.

**Definition 6:**

Origin is defined as the first point \( p_1 \in Tr \) of each trajectory.

**Definition 7:**

Destination is defined as the last point \( p_n \in Tr \) of each trajectory.

**Method**

Our approach extends clustering trajectory proposed by (Cruz et al., 2015) with K-means (Macqueen, 1967) algorithm as follows. Given a set of users, \( U = \{S_1, S_2, \ldots, S_n\} \), where each \( S_i \) is denoted by a tuple set by a user’s trajectory \( Tr_i \) and user’s profile \( P_i \). Optics* generates a set of cluster \( A = \{C_1, C_2, \ldots, C_n\} \), where each
$C_i = \{T_{r_1}, T_{r_2}, \ldots, T_{r_n}\}$ denotes a set of user’s trajectory and it has at least one trajectory from a user called driver $d$ that gives a ride. Next, K-means produces a set of cluster $B = \{X_1, X_2, \ldots, X_n\}$, where each $X_i = \{P_1, P_2, \ldots, P_n\}$ represents a set of user’s profiles. Finally, the set of clusters $A$ and $B$ are combined using an ensemble approach. The result is a set of clusters $R$ of users related profile and trajectory.

Figure 1 describes the entire method. Perceive that the clustering process takes into account proper distances between trajectories and social distance between user’s profile. The method is split into five principal steps: (i) Defining the granularity of user’s trajectory, (ii) Temporal filter, (iii) Optics clustering, (iv) K-means clustering and (v) Relabel and intersection clusters. Since the first three steps are properly described in (Cruz et al., 2015), we present them briefly.

**Trajectory’s Granularity**

Considere $U$ a set of user’s trajectories. Each trajectory contains points that was collected in the short time interval. Because of this, it’s necessary to reduce it. RotaFacil (Telles et al., 2012; 2013) dramatically reduces the number of trajectory’s points by detecting Points of Interest (POIs) (Fig. 2). A new subset $U'$ is thus generated.

**Temporal Filter**

Figure 2 POIs within a circumference for a given radius.

A temporal filter is a way of processing a pipeline in other to find similar trajectories with similar departure and destination times. Surely, it is irrelevant clustering together $T_r(d)$ and $T_r(a)$ if departure and/or destination times of users $d$ and $a$ are very different, even though $T_r(d) \approx T_r(a)$. Regarding $t$ the time of departure ride and $x$ as a bound that user is prepared to accept requests for a ride, the width of the filter is the interval $[t-x, t+x]$. For instance, a user $d$ can offer a ride with departure time at 6 am and inform a boundary of 30 min earlier or later $t$ to accept a request for a ride in an interval of $[5:30, 6:30]$.

**Optics Clustering**

The clustering trajectory is performed by an adaptation of Optics* algorithm (Ankerst et al., 1999). Figure 3 illustrates algorithm’s behavior. $T_r(a)$ belongs to a passenger $a$ who wishes to get a ride whereas $T_r(d)$ belongs to the driver. The similarity only takes into account origin and destination points of the passenger’s trajectory.

---

![Diagram](image)

**Fig. 1:** The method illustrates the step-by-step to reach three types of clusters
Cosine function has been used to compute the similarity (Theodoridis et al., 2010) between two users $\text{sim}(P_i, P_j)$:

$$\text{sim}(P_i, P_j) = \cos(\theta) = \frac{P_i \cdot P_j}{\|P_i\| \|P_j\|}$$  \hspace{1cm} (1)

**Relabel and Intersection**

Relabel strategy is a way to arrange clusters granted similar as exhibited in Fig. 4.

The result obtained by Optics* and K-Means are two partitions (A and B) that are processed by Hugarian Algorithm. These algorithms will have relabeled the partitions to verify which clusters have more users in common. Consider, for instance, that cluster $C_1$ has 4 users with comparable trajectories and cluster $X_2$ has 10 users with similar profiles. If we are in mind that all combinations between A and B, $X_2$ and $C_1$ are clusters that have more users in common.

Figure 5 illustrates relabeling process. The columns and rows represent the partitions and the users respectively. The permutation is used to align the most alike clusters. Users that belong to alike clusters will make part of the final clusters. This work uses trajectory partition as a reference partition which is used as the support to align other partitions. As Fig. 4 and 5 show, the voting approach is not used fully because, in our context, there is no need to vote to generate final clusters with just two partitions.

The consensus functions (represented by $\tau$ on Fig. 4) considers the intersection between clusters:

$$\tau(C_i, X_j) = C_i \cap X_j$$  \hspace{1cm} (2)

because the partition D is resulted from two others partitions: A and B. These final partition is composed by clusters with users that have similar trajectory and profile.
**Experiments**

We have conducted four experiments to prove the feasibility of the ridematching with trajectory and profile. The first two experiments were based on (Cruz et al., 2015). The difference is that we will use a larger real dataset and show the result produced using Optics*. Third experiment shows the results of Optics and K-means using the clusters of user’s profile. Finally, the fourth experiment presents results that were obtained from clusters with users that have both trajectory and profile similarity.

**Datasets**

Three different datasets have been used. The first consists of trajectories of users driving cars or taking buses collected by the Go!Track app. The second consists of trajectories that were reduced by Rota Facil artificially. It was produced 500 trajectories. The third dataset consists of 500 registers of profile attribute also artificially produced.

**GO!Track Dataset**

The Table 1 presents the first dataset. A total of 445 trajectories from 65 different mobile devices were collected between September 2014 and March 2016 in the city of Aracaju/SE. Each trajectory is a set of points obtained at an interval of 0.5 sec (for car) and 10 sec (for bus). The values of these parameters were defined empirically.

Table 2 and 3 show the fields of the dataset. First table stores the collected trajectories and the second one, latitude and longitude mainly.

Figure 6 a trajectory instance and corresponding data.

The present version of the dataset includes an important set of streets and avenues of Aracaju city. We have plotted all the dataset points on Aracaju city map (Fig. 7).

In addition, Table 4 lists the top-20 most visited traffic roads by GO!Track users, according to the number of trajectories (#T) that actually used it. This was accomplished by the Google Geocode API. The column number of points (#P) shows the number of points presented in that traffic road, regardless of the trajectory. We highlight the known principal city traffic roads according to traffic density during peak hours.

Many useful applications use date-time data to predict traffic states or traveling time. Coming graphs provide relations between geographic points and the time. The graph of Fig. 8 presents the similarity between the trajectories and date-time.

The x-axis represents the 163 trajectories and the y-axis the time bands. Each line in the graph represents a trajectory: The higher the line, more time has been spent in the trajectory regardless of the traveled distance. Points represent the city in a diverse range of times, allowing to observe situations where the traffic is probably increased (peak time).
Table 3: Geographic points

| Field   | Description                                      |
|---------|--------------------------------------------------|
| Id      | A unique key to identify each point              |
| Latitude| Latitude from where the point is                 |
| Longitude| Longitude from where the point is               |
| Track_id| The trajectory to which the point belongs        |
| Time    | Date-time the point was collected                |

Table 4: Most visited traffic roads in Aracaju according to GO!Track users

| Order | Address            | #T | #P     |
|-------|--------------------|----|--------|
| 1     | R. Boa Viagem      | 48 | 527    |
| 2     | Av. Pres. Tancredo | 38 | 1068   |
| 3     | Av. Beira Mar      | 34 | 1216   |
| 4     | Av. Ivo do Prado   | 23 | 581    |
| 5     | Av. Mário J. M. Vieira | 22 | 571    |
| 6     | Av. Simeão Sobral  | 21 | 141    |
| 7     | Av. Des. Maynard   | 21 | 950    |
| 8     | BR-235             | 19 | 56     |
| 9     | Av. Confinança     | 18 | 137    |
| 10    | SE-100             | 18 | 280    |
| 11    | Av. Eng. Gentil Tavares | 14 | 253    |
| 12    | Av. Adélia Franco  | 14 | 271    |
| 13    | Av. Filadelfo Dória| 14 | 34     |
| 14    | Av. Dr. José S. R. Filho | 14 | 184    |
| 15    | R. de Muribeca     | 14 | 117    |
| 16    | Av. Barão de Maruim| 13 | 196    |
| 17    | Av. Antônio Cabral | 13 | 123    |
| 18    | Av. Coelho e Campos| 13 | 69     |
| 19    | Av. Farmacêutica C. R | 13 | 158    |
| 20    | Av. João Ribeiro   | 13 | 265    |

Fig. 6: A trajectory instance and corresponding data
Similarly, the graph of Fig. 9 shows the relation between each of the top-20 most visited traffic roads (street and avenues) and the specific instant it was visited. We have considered a daily time interval, in particular, at peak hours. The experiments were realized using the follow parameters adjustment. For experiments one and two the value of the parameter $\varepsilon$ was set to 100, 150, 200 and 300 m. The MinPts was set to 2 and 3. We have assumed that 100 to 300 m are moderate distance limits for a user who wants a ride to move towards the destination point of an offering ride. The values 50, 150, 200 and 250 were set to parameter $\varepsilon$ which is used in cluster extraction algorithm.
In the case of experiments three and four, it was adjustment five attributes: (i) Gender (binary), (ii) reputation (discrete), (iii) age (discrete) \(\text{[child} = 0, \text{teenager} = 1, \text{juvenile} = 2, \text{adult} = 3, \text{elderly} = 4]\), (iv) smoking (binary), (v) music (discrete) \(\text{[poprock} = 0, \text{heavymetal} = 1, \text{classic} = 2, \text{jazz} = 3, \text{reggae} = 4]\). All the attributes have been normalized in order to provide variance 1 and mean 0.

**Evaluation Metrics**

Davies-Boundin Index (DBI) the clustering task. Equation 3 (Theodoridis *et al.*, 2010) was used to evaluate defines the DB value:

\[
DBI = \frac{1}{n} \sum_{i = 1}^{n} \max_{j \neq i} \left( \frac{\sigma_i + \sigma_j}{d(i.e)} \right)
\]

where, \(n\) is the number of clusters, \(c_i\) and \(c_j\) is the centroid of each cluster. The \(\alpha_i\) and \(\alpha_j\) 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 different clusters. Smaller DBI values are better.

**Experimentation Results**

Table 5 shows the results of the first experiment. The radius used in granularity definition step was 25 m. The best result occurs when \(\varepsilon\) was 150 and MinPts were smaller than 3.

Table 6 shows a little contrast compared with Table 5. For example, the best results occur when MinPts was bigger than 2. The radius used in the granularity definition step was 30 m.

Table 7 shows the results of the second experiment. The DBI values among three algorithms are similar. \(\varepsilon\) directly influences clusters’ size according to experimentation. One \(\varepsilon\) that is “big” enough will produce good results. Unlike, small \(\varepsilon\) will generate a plenty of objects with a reachability-distance value equal to undefined. Here, as well as in the work (Cruz *et al.*, 2015) neither method was used to deduce the ideal \(\varepsilon\).

The results of the third experiment are shown in Table 8. The experiment presents a comparison of clustering methods in regard to the user’s profile. First, we show the results for Optics and next for K-means. We can verify that K-means has better results when the number of clusters grows up.

As a consequence of such results, K-means has been used to generate profile clusters in the fourth experiment.

Table 9 shows the results of ensemble learning approach to provide clusters of users with similar trajectories and profiles. The results have been achieved by the matching of trajectory clusters generated by Optics* and profile clusters got with K-means.

Table 9 shows the Number of Final Clusters (NFC), Davies-Boulding Index related to the Trajectory (DBIT) and Davies-Boulding Index related to Profile (DBIP). It shows the results of the Optics* and K-means considering the Number of Clusters (NC) once the ensemble learning approach by DBI metrics is done. DBI metrics is better when a number of clusters are larger (NC ≥ 40).
doesn’t have many user’s trajectories with the similarity less than 200 m or user’s profile aren’t so similar according to values used in this study.

### Complexity Analysis

We have provided some time analysis for our approach. Firstly, we used the software profiling method which is a form of dynamic program analysis. Next, we calculated the time complexity estimation according to big O notation.

The cProfile library enables software profiling analysis. The profiling was used with the purpose to determine which part of the method should be optimized. The analysis covers a set of features such as: A number of function calls NumOfCall, the total time spent by functions or operations, the cumulative time spent by the functions C Time, etc. In addition, we could verify the overall time spent in each method. Table 10 shows the four most expensive functions. Results of Table 10 were obtained with the following setup: \( \epsilon = 100, \) MinPts = 2, \( \epsilon' = 150, \) k = 54.

Table 10 shows that function math.cos has been called a lot of times, but the time spent by the function is less than 10% of neighbors’ function. The neighbor’s function is used by Optics algorithm and has fundamental importance. The neighbor’s function is a bottleneck of Optics algorithm because it consults the whole trajectories’ set when it is called. According to (Ankerst et al., 1999), an index structure like tree-based spatial index can be used and consequently decrease the overall runtime.

The first complexity analysis of the method was done. According to (Ankerst et al., 1999), the Optics* algorithm has an overall runtime of \( O(n^2 \cdot \log n) \) considering a spatial index and similarity algorithm. Additionally, K-means algorithm has an overall runtime of \( O(n^d \cdot \log n) \) where d is the dimension and k the number of clusters. The ensemble approach used has an overall runtime of \( O(K^2) \) considering that Hungarian is been employed. The runtime of ensemble approach can be considered constant because K is a number of the partition that is fixed in 2. Thus, the global runtime is \( O(n^2 \cdot \log n + n^d \cdot \log n) \).

### Conclusion

Encouraging carpooling is an important effort towards the reduction of in-transit vehicles. Although there are some concerned research initiative and even some related software, they do not appropriately treat carpooling context specificities. In this study, we have proposed an extension to the method developed by (Cruz et al., 2015) in order to deal with some of these specificities: Find out groups of users that have similar profile and trajectory and consequently determine potential carpooling possibilities. Furthermore, clustering users’ trajectory and clustering users profile are results that can be considered separately in regard to the final interest of who desire to use the proposed approach as carpooling applications.
Density-based algorithm Optics was chosen due to some features like a minimal number of input parameters, ability to build non-spherical clusters and sturdiness to noise. These features are important to take into account in a task of finding similarity among trajectories. The well-known K-means was chosen because shows better results compared to Optics in a context of users’ profile.

Clustering results and corresponding Davies-Bouldin Index values obtained from a dataset of actual trajectories collected pervasively have shown the feasibility of the proposal. According to experimentation, POIs with different radius seem to not influence the quality of our approach. Initial runtime analysis indicates the elevated complexity. However, if we consider that the problem of finding out similarity among trajectories of users is a special case of the so-called pickup and delivery problem, which is NP-Complete (Agatz et al., 2012), the analysis result proves its feasibility.

We are currently working on experiments that consider weighted profile’s attributes, so the users can define their weighted preferences. We are also currently embedding this similarity approach into a carpooling recommendation service so it could be integrated into some open-source coalition structures to provide proper group’s formation. According to experimentation, POIs with different radius seem to not influence the quality of our approach. Initial runtime analysis indicates the elevated complexity. However, if we consider that the problem of finding out similarity among trajectories of users is a special case of the so-called pickup and delivery problem, which is NP-Complete (Agatz et al., 2012), the analysis result proves its feasibility.

Azzam, E.R. and F.D.D. Bellis, 2008. Carona Brasil. http://www.caronabrasil.com.br/

Acknowledgement

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].

Author’s Contributions

Michael Cruz: Conceptualization, Design, Execution and Analysis drafting

Hendrik Macedo: Conceptualization, Analysis drafting and Critical revision

Adolfo Guimarães: Analysis drafting, Execution and Critical revision

Ethics

There is no ethical issues with this article.

References

Agatz, N., A. Erera, M. Savelsbergh and X. Wang, 2012. Optimization for dynamic ride-sharing: A review. Eur. J. Operational Res., 223: 295-303. DOI: 10.1016/j.ejor.2012.05.028

He, W., X. Wang et al., 2013. Ridesharing: The state-of-the-art and future directions. Trans. Res. Part B: Methodol., 57: 28-46. DOI: 10.1016/j.trb.2013.08.012

He, W., D. Li, T. Zhang, L. An and M. Guo et al., 2012. Mining Regular Routes from GPS Data for Ridesharing Recommendations. Proceedings of the ACM SIGKDD International Workshop on Urban Computing, Aug. 12-12, ACM, China, pp: 79-86. DOI: 10.1145/2346496.2346510

He, W., K. Hwang and D. Li, 2014. Intelligent carpool routing for urban ridesharing by mining GPS trajectories. IEEE Trans Intelligent Transportat. Syst., 15: 2286-2296. DOI: 10.1109/TITS.2014.2315521

Lee, J.G., J. Han and K.Y. Whang, 2007. Trajectory clustering: A partition-and-group framework. Proceedings of the ACM SIGMOD International Conference on Management of Data, Jun, 11-14, ACM, China, pp: 593-604. DOI: 10.1145/1247480.1247546
Levin, I.P., M. Mosell, C. Lamka, B. Savage and M. Gray, 1977. Measurement of psychological factors and their role in travel behavior. Trans. Res. Record, 649: 1-7.

Levy, J., J.J. Buonocore and K. von Stackelberg, 2010. Evaluation of the public health impacts of traffic congestion: A health risk assessment. Environm. Health, 9: 65-65. DOI: 10.1186/1476-069X-9-65

Macedo, H.T., M.O. da Cruz, M.L.S. Matos and A. Guimaraes, 2014. Go!” Registro de Software BR 51 2014 000 963 7, 08 15, 2014.

Macqueen, J., 1967. Some methods for classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, (MSP’ 67), pp: 281-297.

Matos, M.L., M. Cruz, A. Guimarães and H. Macedo, 2014. A social network for carpooling. Proceedings of the 7th Euro American Conference on Telematics and Information Systems, Apri. 02-04, ACM, Chile, pp: 10-10. DOI: 10.1145/2590651.2590662

Mazzella, F., 2004. Blablacar. http://www.blablacar.com

O’Sullivan, S., 2015. Carma. https://carmacarpool.com

Pelekis, N., I. Kopanakis, G. Marketos, I. Ntoutsi and G. Andrienko et al., 2007. Similarity searching trajectories databases. Proceedings of the 14th International Symposium on Temporal Representation and Reasoning, June. 28-30, IEEE Explore Press, Spain. DOI: 10.1109/TIME.2007.59

Resende, P.T.V. and P.R. Sousa, 2009. Mobilidade urbana nas grandes cidades brasileiras: Um estudo sobre os impactos do congestionamento. SIMPOI–SIMPOSIÓ DE ADMINISTRAÇÃO DA PRODUÇÃO, LOGÍSTICA E OPERAÇÕES INTERNACIONAIS, FGV.

Savage, N.S., S. Nishimura, N.E. Chavez and X. Yan, 2010. Frequent trajectory mining on GPS data. Proceedings of the 3rd International Workshop on Location and the Web, Nov. 29-29, ACM, Japan, pp: 3-3. DOI: 10.1145/1899662.1899665

Schrank, D.L. and T.J. Lomax, 2007. The 2007 urban mobility report. Texas Trans. Institute, Texas A & M University, USA.

Shang, S., R. Ding, B. Yuan, K. Xie and K. Zhenge et al., 2012. User oriented trajectory search for trip recommendation. Proceedings of the 15th International Conference on Extending Database Technology, Mar. 27-30, ACM, Germany, pp: 156-167. DOI: 10.1145/2247596.2247616

Telles, R., A.P. Guimarães and H.T. Macedo, 2012. Automated feeding of POI base for the generation of route descriptions. Proceedings of the 6th Euro American Conference on Telematics and Information Systems, May. 23-25, ACM, Spain, pp: 253-259. DOI: 10.1145/2261605.2261643

Telles, R., B. Barroso, A. Guimarães and H. Macedo, 2013. Automatic generation of human-like route descriptions: A corpus-driven approach. J. Emerging Technol. Web Intelligence, 5: 413-423. DOI: 10.4304/jetwi.5.4.413-423

Theodoridis, S., A. Pikrakis, K. Koutroumbas and D. Cavouras, 2010. Introduction to Pattern Recognition: A Matlab Approach. 1st Edn., Academic Press, Burlington, ISBN-10: 0080922759, pp: 231.

Yan, S. and C.Y. Chen, 2011. A model and a solution algorithm for the carpooling problem with pre-matching information. Comput. Industrial Eng., 61: 512-524. DOI: 10.1016/j.cie.2011.04.006

Zimmer, J. and G. Logan, 2012. Lyft. https://www.lyft.com/