Information Systems ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

# Never drive alone: Boosting carpooling with network analysis

R. Guidotti [a,b,*], M. Nanni [b], S. Rinzivillo [b], D. Pedreschi [a], F. Giannotti [b]

[a] KDDLab, Department of Computer Science, University of Pisa, Largo B. Pontecorvo, 3, Pisa, Italy
[b] KDDLab, ISTI-CNR, Via G. Moruzzi, 1, Pisa, Italy

ARTICLE INFO

*Article history:*
Received 30 November 2014
Received in revised form
26 February 2016
Accepted 11 March 2016

*Keywords:*
Carpooling
Complex networks
Rank analysis
Community discovery
Big data analytics
Mobility data mining

ABSTRACT

Carpooling, i.e., the act where two or more travelers share the same car for a common trip, is one of the possibilities brought forward to reduce traffic and its externalities, but experience shows that it is difficult to boost the adoption of carpooling to significant levels. In our study, we analyze the potential impact of carpooling as a collective phenomenon emerging from people's mobility, by *network analytics*. Based on big mobility data from travelers in a given territory, we construct the *network of potential carpooling*, where nodes correspond to the users and links to possible shared trips, and analyze the structural and topological properties of this network, such as network communities and node ranking, to the purpose of highlighting the subpopulations with higher chances to create a carpooling community, and the propensity of users to be either drivers or passengers in a shared car. Our study is anchored to reality thanks to a large mobility dataset, consisting of the complete one-month-long GPS trajectories of approx. 10% circulating cars in Tuscany. We also analyze the aggregated outcome of carpooling by means of empirical simulations, showing how an assignment policy exploiting the network analytic concepts of communities and node rankings minimizes the number of *single occupancy vehicles* observed after carpooling.



## 1. Introduction

There is no need to advocate why traffic and its consequences on the environment, our health and quality of life, and the economy is a major problem for our societies. Carpooling, i.e., the act where two or more travellers share the same car for a common trip, is an old idea brought forward, among many others, to reduce traffic and its externalities. If a large proportion of travellers, especially daily commuters, would adopt carpooling, a substantial traffic reduction could indeed take place. However, experiences from many projects internationally, as we discuss in Section 2, have shown that it is extremely difficult to boost the adoption of carpooling to levels that significantly diminish traffic as a whole. There are many reasons why this happens: psychological, organizational, technological. As a matter of fact, we do not know much yet about the real carpooling potential that emerges from people's mobility—a very preliminary step towards designing the right mechanisms and incentives for a successful carpooling system. Nevertheless, we now have access to the data to observe individual mobility at microscopic level and for large populations of travellers, such as the digitized trajectories of vehicular travels recorded by GPS-enabled on-board devices. These forms of *big data* have been used in [1] to discover the mobility profiles of individual travellers, and to understand when two individuals have compatible matching needs, so that they can share part of their travels. In the present work we

* Corresponding author.
*E-mail addresses:* guidotti@di.unipi.it (R. Guidotti), nanni@isti.cnr.it (M. Nanni), rinzivillo@isti.cnr.it (S. Rinzivillo), pedreschi@di.unipi.it (D. Pedreschi), giannotti@isti.cnr.it (F. Giannotti).

http://dx.doi.org/10.1016/j.is.2016.03.006
0306-4379/

pursue this approach further, to the purpose of understanding the potential impact of carpooling as a collective phenomenon, by adopting a *network analytics* approach. Based on mobility data from a community of travellers in a given territory, we construct the *network of potential carpooling* for that community, where nodes correspond to the users and each link between user $u$ and user $v$ corresponds to the fact that $u$ can take a lift from $v$, because there is a trip in $v$'s profile that can serve $u$ ($u$ can be a passenger of driver $v$). By analyzing the structural and topological properties of this network, we can gain a deeper insight of the potential impact of carpooling. We adapt network analysis tools such as community discovery and node ranking to the purpose of highlighting the subpopulations of travellers that have higher chances to create a carpooling community, and who are the users that show a higher propensity to be either a driver or a passenger in a shared car. Also, we can reason about the propensity of geographical units or cities to carpooling, as well as on the impact on externalities such as $CO_2$ emissions and costs that can be potentially reduced. Our study is anchored to reality thanks to a large mobility dataset, consisting of the complete one-month-long GPS trajectories of more than 150,000 cars observed in Tuscany, the region of central Italy with Florence and Pisa, during the month of May 2011. The population of observed cars is approximately around 10% of all circulating cars. Our analytic observations are therefore referred to real (anonymous) users and real cities, like Florence and Pisa. Remarkably, our method explores the potential of carpooling in *systematic* travels, e.g., home-work commuting, as opposed to ride sharing in occasional trips, which is the approach of several popular apps (see Section 2). Addressing the issue of sharing systematic trips is clearly more challenging and can have a larger impact on traffic reduction. The ultimate contribution of our study is to analyze the potential aggregated outcome of carpooling in the analyzed networks, using several empirical simulations, in terms of expected number of single occupancy vehicles (SOV) that we observe as a result of carpooling matches that take place. We investigate several possible scenarios, and show how a carpooling assignment that exploits the mentioned network analytic concepts of communities and node rankings is the one with the best theoretical performance, because it reduces significantly the expected number of SOV's observed after carpooling. Although much further work is needed to validate in the real world that mining carpooling networks can boost the adoption of ride sharing among communities of commuters, our study is a first in-depth analysis of the potential impact of the approach, which sheds a new, quantitative view on a mechanism that, like all complex social processes, can only be explained in terms of a dynamic network of interacting actors exhibiting an often surprising aggregated behavior.

The rest of this paper is organized as follows. Section 2 contains a detailed overview of related works, addressing carpooling from many different perspectives. The technical background for our study is briefly sketched in Section 3. Section 4 describes the *Never Drive Alone* approach, from the construction of the carpooling network to the assignment method, through the analysis of communities and the ranking measures. Section 5, after illustrating the large mobility dataset used in this study, provides a qualitative and quantitative assessment of the results obtained. Finally, in Section 6, we discuss possible future developments.

## 2. Related work

The carpooling phenomenon is a subject widely studied in the literature. It has been analyzed form various, very different points of view. Carpooling is the second most popular way of commuting, and maybe one of the least understood – a fact that probably explains the need for such a large corpus of studies in the literature.

Carpooling received wide attention in the theoretical literature, mainly regarding high occupancy vehicle lanes (HOV) [2–7]. Refs. [2,4] develop models to calculate the benefits gained for eliminating traffic congestion by adding HOV lanes, or by converting general purpose lanes into HOV. Ref. [5] shows that there is no increase in ridesharing related with the introduction of new HOV lanes, despite the carpooling rate among commuters increases in some periods. Others, like [3,6], consider tolls related with HOV and how these can influence their use. Ref. [7] is a study about carpooling related with the economy world that examines carpooling and driver responses to fuel price changes. It shows that traffic flows in mainline lanes decrease when fuel prices increase, and this effect is stronger when the presence of a HOV lane provides a substitute to driving alone.

Another approach widely followed in the literature for analyzing carpooling is the agent based model (ABM) [8–13]. A multi-ABM in conjunction with Dikstra's algorithm is used in [8] to efficiently answer real time users' queries. In [9] an ABM is designed to optimize transports by the ride sharing of people who usually cover the same route. The information obtained from this simulator are used to study the functioning of the clearing services and the business models. In [10] the authors face the problem by using a multi-ABM to investigate opportunities among simulated commuters and by providing an online matching for those living and working in close areas. Refs. [11,14,13] present a conceptual design of an ABM for the carpooling application to simulate the interactions of autonomous agents and to analyze the effects of changes in factors related to the infrastructure, behavior and cost. They use agent profile and social networks to initiate the ABM, then employ a route matching algorithm and a utility function to trigger the negotiation process between agents. In [12] the authors define an ABM for the individual mobility behavior during carpooling, the criteria and the function to constitute the carpooling community and a protocol for the negotiation of the details of the carpooling trips.

Many carpooling works are related with the study and analysis of mobility data to understand the carpooling phenomena [15–22]. In [15], for example, the authors deeply describe the characteristics of carpoolers, distinguishing among different types of carpooler, and identifying the key differences between a carpooler, a single occupant vehicle (SOV), and a transit commuter. They also describe how and why commuters carpool. In [16], it introduced a methodology for extracting mobility profiles of individuals, and a study criterion to match common routes in order to develop a carpooling service. Something similar is illustrated in [22], which tries to understand mobility patterns, home and work locations, and social ties between users to develop an algorithm for matching users with similar mobility pattern. Ref. [17] proposes a study club model to overtake psychological barriers associated with riding with strangers, to find compatible matches for traditional groups of users and also to find a ride in alternative groups. Using a multilevel regression model and a questionnaire which explains the share of carpooling employees at a workplace, Refs. [18,19] predict the share of carpooling at large workplaces locations, organization and carpooling promotion. In [21] the authors analyze a rail company which provides electric cars to commuters from the home to station trip and then employs the same cars for other works like postal service, medical health care, etc. Finally in [23] the authors develop an application for car sharing recommendation by exploiting a topic clustering algorithm applied to labeled trajectories.

In other studies [24–26,20], the authors try to find simulated or theoretical matches among users asking for a ride in a carpooling scenario and evaluate it in terms of simulated users' feedbacks. Ref. [24] develops and implements the concept of real carpooling by allowing a large base of member passengers and drivers that declared their route to be matched against each other automatically and instantly using mobile phone calls. In [26], the problem is faced as an optimization task reduced to the *chairman assignment problem* [27]. Ref. [28] considers simulated straight-line trajectories observing only origin and destination of trips and classifies users as eligible or ineligible for carpooling by minimizing the time of the trip. In [25] a user network is built that represents planned periodic trips, where the edges are labelled with the probability of negotiation success for carpooling. The probability values are calculated by a learning mechanism using the registered person features, the trip characteristics, and the negotiation feedback. The algorithm provides advice by maximizing the expected value for negotiation success. The differences between the approach proposed in [25] and ours are that we provide matches between couples of users in a pro-active way, suggested from data and not advertised from people. Moreover, Ref. [25] uses the network structure to model the negotiation feedback process, while we use complex networks to model the possible interactions between users and to suggest possible assignments by taking into account real trajectories and systematic movements. Ref. [20] develops a methodology that finds feature points in trajectories and organize them in a tree data structure to speed up and refine geographical queries for carpooling purposes. The authors of [29] introduce a measure of enjoyability based on people's interests, social links, and tendency to connect to people with similar or dissimilar interests and show how this can be used on real world datasets to reduce the number of circulating cars through carpooling by improving at the same time the enjoyability of the trip for both mobility and enjoyability.

Furthermore, there are a few approaches that cannot be clearly assigned to any of the classes discussed above. The work in [30] estimates the energy consumption in terms of fuel related with the impact of casual carpooling. In [31] instead, the authors propose a carpooling based on taxi-cab, that is, they analyze the reduction of circulating taxi in the presence of ride sharing. Moreover, the carpooling problem is investigated also in completely different fields, for instance from the psychological viewpoint [32], and the economical one [33].

Finally, it is worth to note that there are many web sites already operative throughout the world. All of them allow the user to register, search a ride and offer a ride. Anyway, they present several differences.

Drivebook, Roadsharing and Blablacar[1] are some of the most famous ones because they are international, offering intra- and inter-country services. Indeed, they treat mainly long trips. Drivebook is characterized by the feature of being linked with various social networks to improve the confidence among users, while Roadsharing focuses on commuters. The most popular services in the area where our case studies are located (Italy) include Autostradecarpooling, Avacar, Bring-me, Viaggiainsieme, and Autoincomune.[2] Autostradecarpooling, Avacar, and Bring-me are created to find and offer rides for occasional long trips to save money along toll roads and motorways. Viaggiainsieme promotes bike sharing besides routes for commuters. Finally, Autoincomune is mainly oriented towards local mobility, and organizes trips for commuters across neighboring municipalities and also inside the same district.

## 3. Background

In this section we introduce some important concepts that will be useful to follow the rest of the paper. In particular, here we summarize the basics for extracting mobility routines from raw GPS traces, which will be used later to build the network of carpooling opportunities among users; also, we provide some basic definitions related to network analysis, which will be the starting point for computing ad hoc measures for our carpooling networks.

### 3.1. Mobility profiles

Given a set of *users*, their mobility can be described by the set of trips performed in the period of analysis. Each trip, then, is defined by a trajectory, i.e. a sequence of spatio-temporal points:

[1] http://www.drivebook.com/., http://www.roadsharing.com/., http://www.blablacar.com/.

[2] http://www.autostradecarpooling.it, http://www.avacar.it/carpooling/home.aspx, http://www.bring-me.it/, http://www.viaggiainsieme.it/, http://www.autoincomune.it/.

Please cite this article as: R. Guidotti, et al., Never drive alone: Boosting carpooling with network analysis, Information Systems (2016), http://dx.doi.org/10.1016/j.is.2016.03.006

**Definition 1** (*Trajectory*). A *trajectory* $T$ is a sequence of spatio-temporal points $T = \langle (x_1, y_1, t_1), \ldots, (x_n, y_n, t_n) \rangle$, where $x_i$ and $y_i$ $(1 \le i \le n)$ are the coordinates of the $i$-th point and $t_i$ is its corresponding timestamp, with: $\forall 1 \le i < n . t_i < t_{i+1}$.

The set of all the trajectories travelled by a user $u$ makes her *individual history*:

**Definition 2** (*Individual history*). Given a user $u$, we define the *individual history* of the user as the set of trajectories travelled by her and denoted by $H_u = \{T_1, \ldots, T_k\}$.

Using the above definitions and following the profiling procedure proposed in [16], we can retrieve the systematic movements of a certain user $u$, thus inferring directly from mobility data commuting patterns and other routine behaviors of the users. The method consists in clustering the trajectories of the user by means of an ad hoc *distance function* that defines the concept of trajectory similarity to be adopted. In particular, two trajectories closer than a given threshold will be considered similar and contribute to the same mobility behavior:

**Definition 3** (*Trajectory similarity*). Given two trajectories $T$ and $T'$, a trajectory distance function Dist and a distance threshold $\epsilon$, we say that $T$ is *similar* to $T'$ iff $\text{Dist}\,(T, T') \le \epsilon$.

The result of the process is a partitioning of the original dataset of user's trajectories, from which we filter out the *clusters* with few trajectories (statistically non significant behaviors) and the trajectories that are noise (specifically detected by the clustering algorithm). Finally, for each valid cluster remained, we extract a *representative trajectory*, which is called a *routine*. The set of all routines of a user is called her *mobility profile*. More formally:

**Definition 4** (*Routine, mobility profile*). Let $H_u$ be the individual history of a user $u$, $ms$ a minimum size threshold, Dist a distance function and $\epsilon$ a distance threshold. Given a partitioning function $\text{Profile}(H_u, ms, \text{Dist}, \epsilon) = \mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_k\}$, with $H_u \subseteq \bigcup_{i=1}^{k} \mathcal{M}_i$ and $\forall 1 \le i < j \le k . \mathcal{M}_i \cap \mathcal{M}_j = \varnothing$, for each $1 \le i \le k$ we define a *routine* $r_i$ as the medoid trajectory of group $\mathcal{M}_i$. The set of routines extracted from $\mathcal{M}$ is called *mobility profile* and is denoted by $P_u = \{r_1 \ldots r_k\}$. The residual trajectories, i.e. $H_u \backslash \bigcup_{i=1}^{k} \mathcal{M}_i$, represent occasional trips and do not contribute to any routine in the user mobility profile.

Following [16], function Dist will compare trajectories based on their path and on the time of the day they took place. The *mobility profile* of a user describes an abstraction in space and time of her systematic movements: real movements are represented by a set of trajectories describing the generic path followed, and the representative hour of the day it takes place, not instantiated in a specific time and date. Moreover, the exceptional movements are completely ignored due to the fact they will be not part of the profile. Fig. 1 depicts a sample instantiation of the mobility profile extraction process, from the user's trajectories (left) to the clustering represented by function Profile (center) and finally to the resulting routines that form her mobility profile.

### 3.2. Complex network

In this work we will make use of three main concepts belonging to the complex networks field: (i) node degree, (ii) link analysis, (iii) community discovery. Given a directed graph $G$ and one of its nodes $i$, we define the *incoming degree* of $i$ as the number $k_i^{in}$ of links that point to $i$, and its *outgoing degree* as the number $k_i^{out}$ of links that start from $i$ and point to other nodes.

In network science, *link analysis* is a data-analysis technique used to evaluate relationships, i.e. connections, between nodes. In particular we used Hyperlink-Induced Topic Search (*HITS*), also known as *hubs and authorities*, a link analysis algorithm that rates Web pages, developed in [34]. The algorithm assigns two scores to each page: its *authority* score, which estimates the value of the content of the page, and its *hub* score, which estimates the value of its links to other pages. Authority and hub values are defined in terms of one another in a mutual recursion: authority values are computed as the sum of the hub values that point to that page; hub values are the sum of the authority values of the pages it points to.

These hub and authority scores are values that enable us to rank nodes according to some criteria. We define *HITS* as a *ranking function*:

**Definition 5** (*Ranking measure*). Given a direct graph $G = \langle N, E \rangle$, we define the ranking function ranking($G$) as the algorithm *HITS*, taking as input $G$ and returning two score vectors $h$ and $a$, respectively for *hub* and *authority*.

Finally, *community discovery* is the problem of identifying communities hidden within the structure of a complex network [35]. A *community* is a set of entities that, in the network sense, are closer to the other entities of the community than with those outside it. Thus, communities are groups of entities that share some common properties and/or play similar roles. In the literature, several popular community discovery algorithms exist [36–38]. Among them, in this work we choose to adopt *Infohiermap* for its ability to deal with direct graphs and for the efficient ranking random surf approach it implements.

**Definition 6** (*Community discovery*). Given a direct graph $G = \langle N, E \rangle$, we define the function communities($G$) as the algorithm *Infohiermap*, taking as input $G$ and returning a set of communities $\mathcal{C} = \{C_1 \ldots C_n\}$, where $C_i \subseteq N$ is a set of nodes.

## 4. Never drive alone

In this section we describe an approach for realizing a carpooling service, based on the identification of pairs of

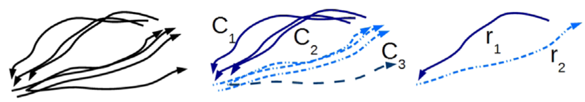



**Fig. 1.** The user's *individual history* (left: black lines), the clusters identified by the grouping function (center: $C_1, C_2, C_3$) and the extracted *individual routines* (right: $r_1, r_2$) forming her *individual mobility profile*.

Please cite this article as: R. Guidotti, et al., Never drive alone: Boosting carpooling with network analysis, Information Systems (2016), http://dx.doi.org/10.1016/j.is.2016.03.006

users that could share their vehicle for one or more of their systematic trips. The method builds on and develops several of the concepts summarized in the previous section.

In the following we propose a procedure for suggesting carpooling assignments – i.e. offering to some users to become a driver for other users, who will become their passengers – among systematic users. The output of such procedure also provides the means for studying the potential of carpooling on the area of analysis. The procedure is composed by two main tasks. The first one regards the construction of the carpooling network, the calculus of the ranking scores and the extraction of the communities. The second one concerns the actual assignment of drivers and passengers among the users that form the carpooling network, exploiting the ranking score and the community information computed before.

### 4.1. Carpooling network construction

We talk about *carpooling interaction* when a user can get or offer a ride to another one. The idea is to use complex networks to model the potential carpooling interactions, to use the ranking measures to evaluate how much a user is suitable for being a driver or a passenger, and to use community detection to characterize groups of users that are highly related in terms of carpooling.

The starting point of this analysis is the set of routines which constitutes the user mobility profiles. Since mobility profiles represent users' systematic behaviors, by comparing them it is possible to understand if a user can be served by another one. The system can keep reasonably up-to-date routines and profiles by executing the profiling process regularly, for instance every week, over the most recent mobility data.

A basic operation we need to perform is testing whether a routine is *contained* in another. If a routine $r_1$ is contained in a routine $r_2$ then the user that systematically follows $r_1$ could leave her car at home and travel with the user that systematically follows $r_2$. The relation of routine containment is defined as follows:

**Definition 7** (*Routine containment*). Given two routines $r_1 = \{(x_1^{(1)}, y_1^{(1)}, t_1^{(1)}), \ldots, (x_n^{(1)}, y_n^{(1)}, t_n^{(1)})\}$ and $r_2 = \{(x_1^{(2)}, y_1^{(2)}, t_1^{(2)}), \ldots, (x_m^{(2)}, y_m^{(2)}, t_m^{(2)})\}$, a *spatial tolerance* $spat_{tol}$ and a *temporal tolerance* $temp_{tol}$, we say that $r_1$ *is contained in* $r_2$, i.e. $\text{contained}(r_1, r_2, spat_{tol}, temp_{tol})$, if $\exists\, i, j. 1 \le i < j \le m$ such that:

$$\|(x_1^{(1)}, y_1^{(1)}) - (x_i^{(2)}, y_i^{(2)})\| + \|(x_n^{(1)}, y_n^{(1)}) - (x_j^{(2)}, y_j^{(2)})\| \le spat_{tol} \wedge$$

$$|t_1^{(1)} - t_i^{(2)}| + |t_n^{(1)} - t_j^{(2)}| \le temp_{tol}$$

where

- $spat_{tol}$ is the maximum total distance that the user which *is served* could walk to reach the pick-up point, and to reach her final destination from the get-off point;
- $temp_{tol}$ is the maximum total amount of time that the user which *is served* is allowed to waste, as delay or anticipation w.r.t. her original trip, considering the departure and the arrival time.

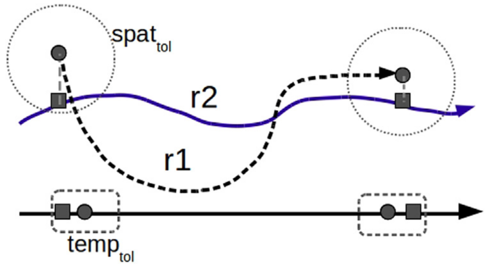



**Fig. 2.** Example of routines containment: $r_1$ *is contained* in $r_2$ because the starting and ending points of $r_1$ (circular points) are spatially and temporally close enough to some points of $r_2$ (squared points).

It is important to note that the *contained* relation is not symmetric, since one routine might include another without having the vice versa holding. This can happen when the routines compared have different lengths, in which case the origin of the user which *serves* the other can be very far from the origin of the one who is *served*, and similarly for the destination point. Fig. 2 provides a visual depiction of the containment relation over a simple example. This formulation basically assumes that the users *served* (i.e. the candidate passengers) are willing to walk and change their time schedule in exchange of the ride they get, while the users which *serve* (i.e. the candidate drivers) do not change their routine.

Using the routine containment relation it is possible to build a *carpooling network* $G = \langle N, E \rangle$. Given a set of profiles $\mathcal{P} = \{P_1, \ldots P_n\}$, for each pair of different users $u$ and $v$, we check the routine containment between every routine $r_i^u \in P_u$ and every routine $r_j^v \in P_v$. If $contained(r_i^u, r_j^v, spat_{tol}, temp_{tol})$ holds, then $u, v \in N$ and $\{(u, v, r_i^u, r_j^v)\} \in E$.

**Definition 8** (*Carpooling network*). A *carpooling network* $G = \langle N, E \rangle$ is a multi-dimensional graph where $N$ represents the set of all users taking part in at least a carpooling interaction, $E$ is the set of all labeled edges $(u, v, r_i^u, r_j^v)$, where $r_i^u$ is a routine of $u \in N$, $r_j^v$ is a routine of $v \in N$, and $r_i^u$ is *contained* in $r_j^v$.

Note that the *carpooling network* guarantees that the trajectories considered are routines, and therefore they are repeated systematically ensuring that a ride is most likely available or needed on that route. In Fig. 3 (left) we have a representation of the carpooling network. Given a carpooling network $G = \langle N, E \rangle$ we define the *possible passengers* and *possible drivers* as follows:

**Definition 9** (*Possible passengers*). Given a carpooling network $G = (N, E)$, a user $u \in N$ is a *possible passenger* if she has at least an outgoing link, that is $k_u^{out} > 0$.

**Definition 10** (*Possible drivers*). Given a carpooling network $G = (N, E)$, a user $u \in N$ is a *possible driver* if she has at least an in-going link, that is $k_u^{in} > 0$.

We denote with $PP_G$ the set of all possible passengers and with $PD_G$ the set of all possible drivers in $G$. Note that it is possible (and actually rather frequent) that $PP_G \cap PD_G \neq \emptyset$, thus some user can act both as possible passenger and possible driver.

Please cite this article as: R. Guidotti, et al., Never drive alone: Boosting carpooling with network analysis, Information Systems (2016), http://dx.doi.org/10.1016/j.is.2016.03.006

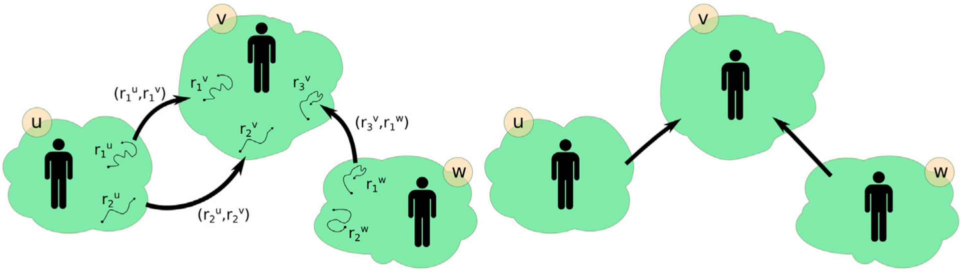



**Fig. 3.** Carpooling network (left) and carpooling user network (right).

Finally, it is worth to highlight that a carpooling network is in fact a multidimensional network: users $u$ and $v$ can share for example two routines; the going trip and the return trip because they take place at different times and also on different roads. However, in order to use some common network analytic tools we have to transform the carpooling network in a mono-dimensional network (see Fig. 3 (right)).

**Definition 11** (*Carpooling user network*). Given a carpooling network $G = \langle N, E \rangle$, we define a *carpooling user network* as a direct mono-dimensional graph $G' = \langle N, E' \rangle$ obtained by collapsing all multi-dimensional edges between the same pair of users, i.e. $E' = \{(u,v) | (u,v,r_1^u,r_1^v) \in E\}$.

Since $G'$ is a direct network, then an arc $(u,v)$ is directed from $u$ to $v$, consequently $v$ is said to be a *successor* of $u$.

### 4.2. Greedy carpooling assignment suggestion

Using the carpooling network, we are now able to extract potential assignments. The *carpooling assignment* method proposed in this section follows a simple heuristic and a greedy idea. The method takes as input a carpooling user graph $G$, i.e. multidimensional edges are not considered, assuming that each pair of users can share only one routine: the general case will be described later as an extension of the solution described here. The idea is that this first procedure is applied to a relatively short time window within the day, where it is basically certain that each user will have at most one *active* routine, e.g. in a typical situation a time window covering the period from 8 a.m. to 8:15 a.m. might contain the home-to-work routine of a commuter, but not the symmetric one, which will likely appear in another time slot in the afternoon. In Section 4.4 we will describe the overall algorithm that iteratively applies the present one on different time slots. The output of the method is a classification of the users taking part in the carpooling network. In particular, the set $D$ contains the drivers that host some passengers in their car, $P$ contains the passengers that are hosted by some drivers, and $S$ contains the single-occupant-vehicles (SOV) that drive alone. The three classes form a partitioning of the users, i.e. $N = D \cup P \cup SOV$ and $|N| = |D| + |P| + |SOV|$.

Algorithm 1 illustrates the *greedy assignment* method. This procedure uses a sorting function $f$ to order the *possible passengers* $u$ according to some criteria $c'$ and it iterates through them (*line* 1). If $u$ has not been already assigned as driver or passenger (*line* 2), the algorithm iterates through her *possible drivers* $v$ (i.e. the out-linked nodes in the network, the function successors($u$) returns the set of successors of $u$) using $f$ according to another criteria $c''$ (*line* 3). If $v$ has free places in her car (*line* 4), then $u$ is assigned as passenger (*line* 6) and $v$ as driver (*line* 5). The procedure is repeated until every user is assigned, or there are no free places left. All the nodes that have not been assigned as driver or passenger, then they are considered SOV (*lines* 12–16).

We remark that the algorithm is intended to be applied iteratively on successive time windows, therefore it takes as input also the output sets obtained from previous iterations, in order to consider in the matching process all users that are not already and completely assigned. For example, if a driver has already used all her free places for an active routine, then she cannot take other passengers, and therefore she is not considered in the matching at the present iteration. On the other hand, a user that was classified as SOV for an active routine can still be considered both as possible passenger and possible driver.

The main purpose of this procedure is to reduce the number $|S|$ of systematic cars in which the driver is driving alone and, in second instance, the total number of systematic cars in circulation given by $|D| + |S|$, thus increasing the number of systematic cars that are not needed anymore – corresponding to the number of users that turned into passengers, $|P|$. The most important component is represented by $|S|$, since SOVs do not play an active role in carpooling although they could potentially share at least one routine with another user – a basic prerequisite for being part of the network. The algorithm is parametric with respect to the sorting criteria used. As baseline sorting criteria we adopted a random sorting, that is, the nodes are ordered in a random way. Other, more sophisticated criteria are discussed in Section 4.5.

Although the algorithm has a quadratic complexity, in practical cases it is essentially linear in the number of nodes analyzed, $O(|N|)$. This happens because if a node has already been marked as driver or passenger, then it cannot be re-analyzed. Also the presence of a inner loop does not lead to quadratic complexity because this would mean that every possible driver could offer a lift to all (or a large part of) possible passengers, which is highly improbable. Moreover, we have to consider the cost of the sorting functions $f$, which is $\Theta(N \log N)$ in the worst case. The cost of the innermost sorting function could be at worst $\Theta(N^2 \log N)$ but, as above, this would happen if every node links to all the others. In practice, the innermost sorting function $f$ function cost is $O(k_u^{out} \log k_u^{out})$ each time it is repeated, i.e. $O(Nk_u^{out} \log k_u^{out})$. Since the average $k_u^{out}$ is very low in this kind of networks, we have that $O(k_u^{out} \log k_u^{out})$ can be approximated to a constant $c$. Thus, the dominant cost remains $\Theta(N \log N)$.

**Algorithm 1.** *calculateGreedyAssignment* $(G', f, m, c', c'', D, P, S)$.

**Input** : $G' = \langle N, E \rangle$ - carpooling user network, $c', c''$ - sorting criteria, $f$ - sorting function, $m$ - max number of free places, $D$ - set of sets of possible driver containing the assigned passengers (e.g. $D_v$ is the set of passengers assigned to driver $v$), $P$ - set of sets of possible passengers containing the assigned driver (e.g. if $v \in P_u$ it means that passenger $u$ is assigned to driver $v$, $|P_u| \le 1$ always) $S$ - set of single occupant vehicle

**Output**: $D, P, S$

```
for u ∈ f(N, c′) do
    if D_u ⊈ D ∧ P_u ⊈ P then
        for v ∈ f(successors(u), c″) do
            if |D_v| ≤ m then
                D_v ← D_v ∪ {u};
                P_u ← {v};
                break;
            end
        end
    end
end
for u ∈ N do
    if D_u ⊈ D ∧ P_u ⊈ P then
        S ← S ∪ {u};
    end
end
return D, P, S;
```

The problem analyzed is NP-complete [39], and an optimal approach to solve it is exponential in the number of edges. Indeed, such an approach should take into account the fact that every assignment might inhibit any of the others, since each node in the network can either be a driver or a passenger and once the choice is made it cannot be reversed, then virtually all combinations must be tried in order to find the best one. Finally, we note that, in spite of its resemblance with bipartite matching, our formulation of the carpooling problem cannot be solved just using a maximal matching over the bipartite graph among possible drivers and possible passengers, because the intersection between possible drivers and possible passengers is not empty. Thus, in order to reduce it to the bipartite case we should evaluate the matching over all its possible bipartite projections, i.e. by assigning all users to one fixed role, trying all possible combinations. That is computationally equivalent to the exhaustive, brute force approach mentioned above. For these reasons, the solution we propose is an heuristics, which trades optimality for scalability.

### 4.3. *Ranking criteria and problem partitioning*

In order to find the best assignments among the users taking part in the carpooling scenario, it is useful to discover the best passengers and the best drivers among the candidate ones. We say that a user is a “good passenger” if she can accept a lift from many “good drivers”, and mutually, a user is a “good driver” if she can offer a ride to many “good passengers”. Thus, we analyze the carpooling network to rank a user as a “good passenger” or as a “good driver”. The idea to reach this goal is to consider the carpooling user graph and the apply the *HITS* algorithm [34]. Indeed, the HITS task of extracting hub and authority scores to estimate the value of a web page can be directly mapped to the carpooling scenario for measuring how much a user is suitable for being a good passenger or a good driver. In the context of carpooling networks, we

Please cite this article as: R. Guidotti, et al., Never drive alone: Boosting carpooling with network analysis, Information Systems (2016), http://dx.doi.org/10.1016/j.is.2016.03.006

define the hub score as *passengerness*, i.e. the attitude of $u$ for being a good passenger, and the authority score as *driverness*, i.e. the attitude of $u$ for being a good driver.

**Definition 12** (*Passengerness and driverness*). Given the carpooling user network $G = \langle N, E \rangle$ and its adjacency matrix $A$, for each user $u \in N$, we define *passengerness* $p_u$ and *driverness* $d_u$ respectively as the hub and authority scores of $u$ in $G$. Formally, vectors $p$ and $d$ are eigenvectors such that $p = AA^T p$ and $d = A^T Ad$.

Even though the *passengerness* and the *driverness* are indicators of how much a user can be a good driver or a good passenger, they do not provide information about which groups of users could more easily travel together, or which geographical areas could be more promising for a carpooling service. Consequently, we extracted groups of users sharing common routines, which have then been analyzed to characterize each group geographically (to understand whether such groups are localized or dispersed over large areas), and with respect to their *passengerness* and *driverness*.

**Definition 13** (*Carpooling community*). Given a carpooling user network $G' = \langle N, E' \rangle$ we define a *carpooling community* $C \subseteq N$ as a group of users who share more routines with the users inside the community rather than with the users outside the community.

In order to extract the carpooling communities and to perform the carpooling suggestions without discarding the temporal knowledge we introduce carpooling temporal networks:

**Definition 14** (*Carpooling temporal network*). Given a multi-dimensional carpooling network $G = \langle N, E \rangle$, a time stamp *ts* and a temporal duration *dur*, we define a *carpooling temporal network* as a direct graph $G' = \langle N', E' \rangle$ such that $E' = \{e_{uv} \in E | isActive(e_{uv}, ts, dur)\}$ and $N' \subseteq N$ is the set of all nodes comparing in $E'$. The isActive operator is defined as

$$\text{isActive}(e_{uv}, ts, dur) \equiv (ts \le t_1^{r_i} < ts + dur) \wedge (ts \le t_n^{r_i} < ts + dur)$$

where $t_1^{r_i}$ is the time stamp of the first point of $r_i$ and $l = t_n^{r_i}$ is the time stamp of the last point of $r_i$ (Fig. 4).

An edge $e_{uv}$ is active if the contained routine is not finished in a certain time window. Note that a *carpooling temporal network* is a mono-dimensional direct graph if the used time window is short enough (i.e., *dur* is relatively small) and there are not two users $u$ and $v$ that systematically follow two different pairs of matching routines in the same time window – usually a rather extreme phenomenon for reasonable values of *dur*. A *carpooling network* can be seen as a particular *carpooling temporal network* where every edge is active. Finally, we highlight that a *carpooling temporal network* is different from a *carpooling user network*, since the second considers every carpooling interaction.

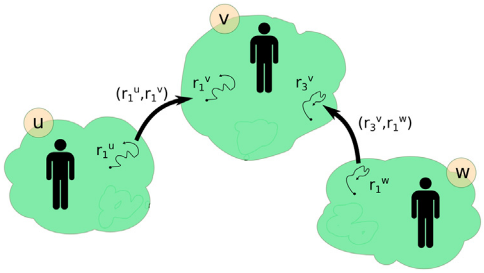



**Fig. 4.** Carpooling temporal network.

## 4.4. Never drive alone method

Using the measures and concepts defined up to now, we describe in the following the *Never Drive Alone* (NDA) method which tries to minimize the number of SOVs. The detailed procedure is described in Algorithms 2 and 3. The main difference between these two versions is that the second one uses the community information, while the first one does not. NDA performs the following steps: (i) extracts the systematic movements (*lines* 3–6); (ii) builds the carpooling network (*lines* 7–8); (iii) calculates the *passengerness* and *driverness* ranking scores (*line* 10); (iv) extracts the carpooling communities (*lines* 11); (v) makes the assignments and classify the users as drivers, passengers or SOVs using Algorithm 1 (*lines* 14–*end*).

Given a time window defined by the parameters *ts* and *dur* discussed in the previous section, function removeFinishedInteractions removes from $D', P', S'$ the assignments that will not be active in the next time window because they end in the current one. In this way, a driver can offer a lift to more then $m$ (max number of free places) users because if she systematically travels a long routine, she might drop-off a passenger and later take another one, also multiple times. The returned sets classify the user according to their role in the carpooling scenario. That is, a user will be in $S$ if and only if she is left out from every carpooling interaction in every time window. If a user can physically act either as a driver or as a passenger then she is counted as a driver because for at least a systematic trip she offered a ride and thus used her car. This happens for example when a user offers a ride to someone in the morning, then returns to the starting point and finally in the afternoon takes a lift to go somewhere else.

When the procedure is performed taking into account the carpooling communities (see Algorithm 3), for each time stamp considered the communities are extracted and analyzed in a certain order which can depend on the size of the community. The purpose is to reduce the focus assignment problem on sets of users that are similar in the carpooling sense, that is, we give to the edges of nodes belonging to different communities a lower importance, because they are expected to offer a ride or get a lift with lower probability – typically because different communities often correspond to different geographical areas. On the contrary, users in the same communities are similar each other, thus their links are evaluated with an high importance in suggesting assignments.

**Algorithm 2.** *NeverDriveAlone* ($\mathcal{H}$, *dur*, *f*, *m*).

**Input** : $\mathcal{H}$ - dataset of user movements, $ts$ - start of time window, $dur$ - temporal duration, $f$ - sorting function, $m$ - max number of free places,
**Output**: $D$ - set of drivers, $P$ - set of passengers, $S$ - set of SOVs

$D \leftarrow \emptyset$; $P \leftarrow \emptyset$; $S \leftarrow \emptyset$;
$\mathcal{P} \leftarrow \emptyset$; `/* set of profiles */`
**for** $H_u \in \mathcal{H}$ **do**
  $Pr_u \leftarrow \text{Profile}(H_u)$; `/* extracts user mobility profile */`
  $\mathcal{P} \leftarrow \mathcal{P} \cup Pr_u$;
**end**
$G \leftarrow \text{buildCarpoolingNetwork}(\mathcal{P}, \textit{contained}(*))$;
$G' \leftarrow \text{extractCarpoolingUserNetwork}(G)$;
$k^{out}, k^{in} \leftarrow \text{getDegrees}(G')$; `/* calculates out-degree and in-degree values */`
$p, d \leftarrow \text{ranking}(G')$; `/* calculates passangerness and driverness using HITS */`
$c' \leftarrow \text{createSortingCriteria}\,(k^{out}, p)$; `/* creates the first sorting criteria */`
$c'' \leftarrow \text{createSortingCriteria}\,(k^{in}, d)$; `/* creates the second sorting criteria */`
$D' \leftarrow \emptyset$; $P' \leftarrow \emptyset$; $S' \leftarrow \emptyset$;
**for** *selected* $ts$ **do**
  $G^{ts,ts+dur} \leftarrow \text{extractCarpoolingTemporalNetwork}(G, ts, dur)$;
  $D', P', S' \leftarrow \text{calculateGeedyAssignment}(G^{ts,ts+dur}, f, m, c', c'', D', P', S')$;
  $D, P, S \leftarrow \text{updateAssignments}(D, P, S, D', P', D')$;
  $D', P', S' \leftarrow \text{removeFinishedInteractions}(G^{ts,ts+dur}, D', P', S', ts, dur)$;
**end**
**return** $D, P, S$;

**Algorithm 3.** *NeverDriveAloneCommunities* ($\mathcal{H}$, *dur*, *f*, *m*).

**Input** : $\mathcal{H}$ - dataset of user movements, $ts$ - start of time window, $dur$ - temporal duration, $f$ - sorting function, $m$ - max number of free places,
**Output**: $D$ - set of drivers, $P$ - set of passengers, $S$ - set of SOVs

$D \leftarrow \emptyset$; $P \leftarrow \emptyset$; $S \leftarrow \emptyset$;
$\mathcal{P} \leftarrow \emptyset$; `/* set of profiles */`
**for** $H_u \in \mathcal{H}$ **do**
  $Pr_u \leftarrow \text{Profile}(H_u)$; `/* extracts user mobility profile */`
  $\mathcal{P} \leftarrow \mathcal{P} \cup Pr_u$;
**end**
$G \leftarrow \text{buildCarpoolingNetwork}(\mathcal{P}, \textit{contained}(*))$;
$G' \leftarrow \text{extractCarpoolingUserNetwork}(G)$;
$k^{out}, k^{in} \leftarrow \text{getDegrees}(G')$; `/* calculates out-degree and in-degree values */`
$p, d \leftarrow \text{ranking}(G')$; `/* calculates passangerness and driverness using HITS */`
$\mathcal{C} \leftarrow \text{communities}(G')$; `/* extracts the users' communities */`
$c' \leftarrow \text{createSortingCriteria}\,(k^{out}, p)$; `/* creates the first sorting criteria */`
$c'' \leftarrow \text{createSortingCriteria}\,(k^{in}, d)$; `/* creates the second sorting criteria */`
$D' \leftarrow \emptyset$; $P' \leftarrow \emptyset$; $S' \leftarrow \emptyset$;
**for** *selected* $ts$ **do**
  $G^{ts,ts+dur} \leftarrow \text{extractCarpoolingTemporalNetwork}(G, ts, dur)$;
  **for** $C \in \mathcal{C}$ **do**
    $G_C^{ts,ts+dur} \leftarrow \text{extractSubGraph}(G^{ts,ts+dur}, C)$;
    $D', P', S' \leftarrow$ $\text{calculateGeedyAssignment}(G_C^{ts,ts+dur}, f, m, c', c'', D', P', S')$;
  **end**
  $D, P, S \leftarrow \text{updateAssignments}(D, P, S, D', P', D')$;
  $D', P', S' \leftarrow \text{removeFinishedInteractions}(G^{ts,ts+dur}, D', P', S', ts, dur)$;
**end**
**return** $D, P, S$;

Please cite this article as: R. Guidotti, et al., Never drive alone: Boosting carpooling with network analysis, Information Systems (2016), http://dx.doi.org/10.1016/j.is.2016.03.006

### 4.5. Sorting and matching strategies

Both Algorithms 2 and 3 rely on the greedy procedure reported in Algorithm 1. It is worth to underline that this procedure is based on the knowledge extracted form data. Indeed, the structure of the greedy assignment exploits the fact that the carpooling networks show a power low distribution of the nodes' degree (see the detailed study provided in Section 5.2). By using smart sorting criteria, our purpose is to lead the algorithm to consider first the least "promising" passengers (i.e. the most difficult ones to match), and then by ordering their drivers, to assign the worst passengers with their least promising drivers. This way, passengers with less possibilities to be matched are assigned first, while passengers which have more opportunities are assigned to the remaining drivers. We can instantiate this reasoning both using the in/out degrees and using the passengerness/driverness ranking criteria.

In this work we consider the following criteria, in order of complexity:

- (*r*) *random criteria* ($c' = \{random$ order$\}, c'' = \{random$ order$\}$): users are sorted randomly both if they are drivers or passengers;
- ($g_1$) *degree criteria* ($c' = \{k^{out}$ ascending order$\}, c'' = \{k^{in}$ ascending order$\}$): users are sorted according to the carpooling user network *out-degree* $k^{out}$ and *in-degree* $k^{in}$, that is, the nodes are sorted by increasing $k^{out}$ and than, their neighbors are ordered by increasing $k^{in}$;
- ($g_2$) *degree – ranking scores criteria* ($c' = \{(k^{out}, p)$order$\}$, $c'' = \{(k^{in}, d)$order$\}$): users are sorted according to *passengerness* $p$ and *driverness* $d$ in addition to $k^{out}$ and $k^{in}$, that is, the nodes are sorted in a lexicographical order by increasing $(k^{out}, p)$ and then, their neighbors are sorted in a lexicographical order by increasing $(k^{in}, d)$.

In principle, the methodology can be applied also switching passengers with drivers, i.e. by enumerating drivers first, and then matching each of them with her possible passengers. Yet, preliminary experiments proved that this order is largely less successful than the original one presented above. Therefore, in the rest of the paper we will consider only the passengers-first approach.

Another information that can be exploited to guide NDA is the community membership. Therefore, we consider two further variants of the method: a basic one, which is agnostic of the communities; and a community-driven one, where the matches between intra-community individuals have priority over all the others:

- *(w) plain version*, Algorithm 2, considering every edge in the whole network with the same importance;
- *(c) prioritized version*, Algorithm 3, that suggests an assignment to the users inside the same community and then, if that fails, among users of different communities.

Finally, we adopted two strategies for considering the temporal dimension. The mobility profiles and the *contained* function for comparing any pair of profiles make the carpooling network basically a summary of a typical day made of systematic routines and their mutual inclusion relations. We can decompose this day in a series of time slots with a predefined duration (*dur*), obtaining a series of carpooling temporal networks. The way the sequence of time slots is produced is a parameter of the general method. Here we consider two main variants:

- *(discrete)* time slots, they start at discrete time instants, for instance one every 5 min starting from midnight. This produces a sliding window of length *dur* that moves of step 5 min;
- *(continuous)* the time slots, they start in correspondence of the last successful carpooling interaction, i.e. the time of the last matched routines becomes the next starting time.

In Section 5.3 we will evaluate experimentally each combination of the three parameters discussed here (sorting criterion, usage of communities, choice of time slots).

## 5. Impact on real mobility

In this section we illustrate an instantiation of the overall approach proposed to a real case study, and show the results obtained. The section is divided into three main parts, corresponding to the phases of the methodology proposed: extraction of user profiles, construction of a car pooling network, and selection of carpooling suggestions. Also, a sub-section on computation times is provided, together with a summary of the main results presented.

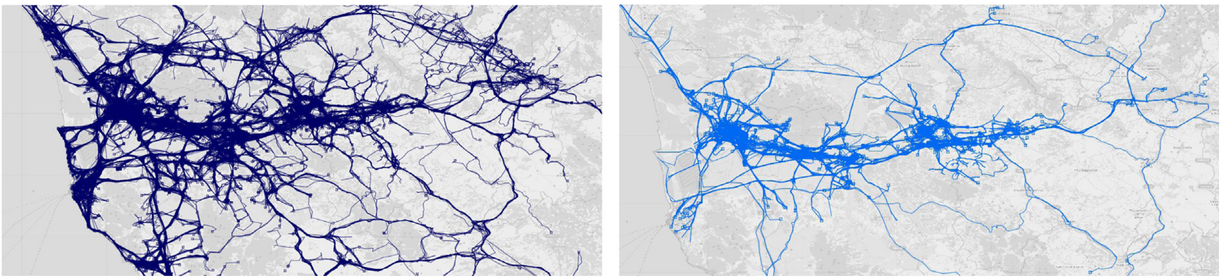


**Fig. 5.** (Left) A sample of trajectories in Pisa province. (Right) The mobility profiles extracted.

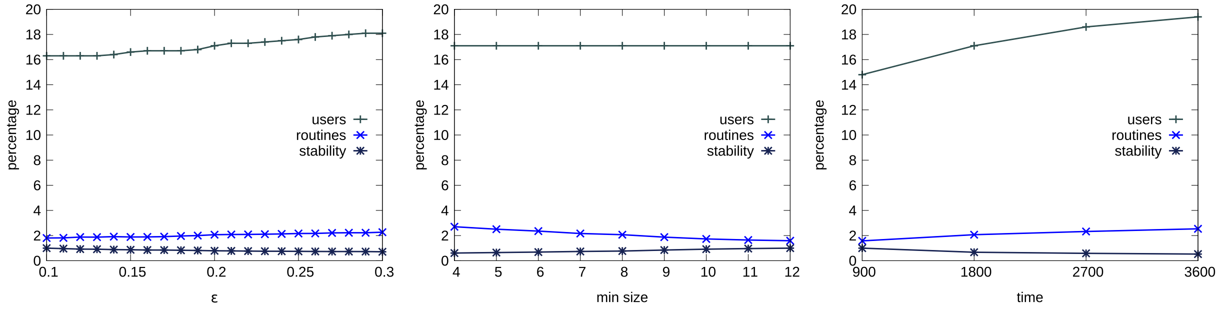



**Fig. 6.** Profile test parameters eps (left), min size (center), and time threshold (right).

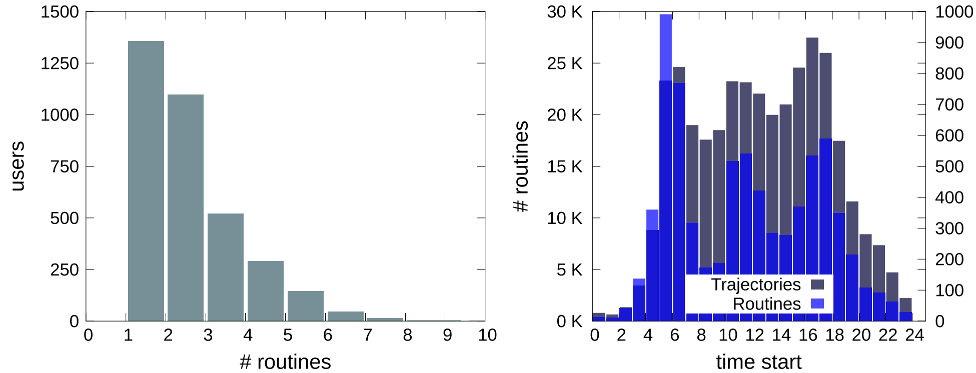



**Fig. 7.** Distribution of routines per user (left) and starting time of trajectories and routines (right).

## 5.1. User profiles

*Dataset*: As a proxy of human mobility, we used real GPS traces collected for insurance purposes by *Octo Telematics S.p.A* [40]. The full dataset contains 9.8 million car travels performed by about 159,000 vehicles active in a geographical area focused on Tuscany over a period from 1 to 31 May 2011. Fig. 5 (left) depicts a sample of the considered trajectories.

Since the area and the period described in our mobility dataset contain an heterogeneous mix of contexts and conditions, in order to obtain better interpretable results we split the data along time and geography. On the temporal dimension, we separated working days and non-working days, since it is commonly observed that during Saturday and Sunday most people leave their working mobility routines and adopt other more erratic behaviors. Moreover, given our focus on systematic mobility, we filtered out weekend trajectories maintaining only those from Monday to Friday of every week. In order to consider also the heterogeneity of the territory covered by the dataset, we split it into provinces, each containing all the trajectories that pass through it. In particular, in this work we are reporting the results obtained for Pisa and Florence provinces, which represent two rather different kinds of mobility, both in terms of population and traffic flows. Finally, too short trips (less than 1 km) have been removed.

*Profiles construction*: Since the starting point of our methodology are users' individual routines, we began with a set of tests aimed to retrieve the best parameters to extract reliable mobility profiles. The clustering algorithm used to extract the routines is a variant of *OPTICS*, a density-based clustering algorithm [41], which thus constitutes our grouping function used in Definition 4. In OPTICS, we employed the same distance function used in [16] for the clustering step. To tune the parameters, we have studied OPTICS' settings on a subset of 1000 users in the Pisa dataset. In particular, we studied three parameters: $\epsilon$, *min size* and *time*. $\epsilon$ was varied in the range $[0.1, 0.3]$ with step 0.01, Fig. 6 (left). The bigger the $\epsilon$, the more different trajectories are clustered together. In other terms, it expresses the similarity required between trajectories. Parameter *min size* was varied in the range $[4, 12]$, Fig. 6 (center). It represents the minimum number of trajectories that must be in a cluster to be considered valid. Finally, we observed the time threshold *time* varying in $\{900, 1800, 2700, 3600\}$ s, see Fig. 6 (right). It is the maximum tolerated difference between the starting times of two trajectories, and is used by the clustering function to decide if two trajectories are well synchronized. The criteria we consider to tune the values are: (i) the dataset coverage, in terms of users having routines, (ii) the average number of profiles per user, and (iii) the stability of profiles, i.e. the number of profiles per user that are consistently preserved along the whole duration of the dataset. The resulting plots do not show particular breaking points nor strong trends, thus suggesting that the choice of this parameters is not critical. Yet, we can observe some minor change of the distributions around the middle value of each figure, e.g. the *time* curves change more rapidly after 1800 s. Therefore, we choose $\epsilon = 0.2$ (approximatively

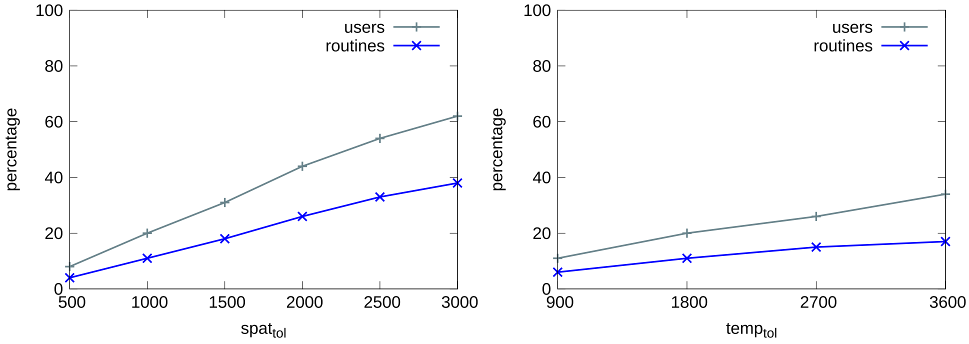



**Fig. 8.** Network construction test on the parameters of *contained*: (left) $spat_{tol}$ and (right) $temp_{tol}$.

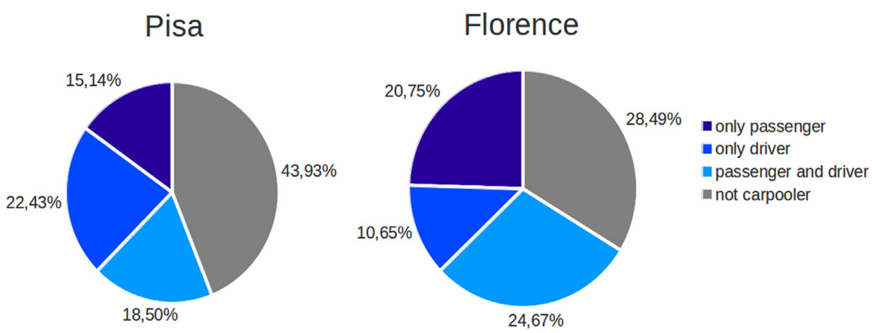



**Fig. 9.** Carpoolers classification pie chart for Pisa and Florence.

meaning a 80% of similarity), *time*=30 min (i.e. 1800 s) and *min size* =8 (i.e. all trips repeated at least 8 times over our 20 working days will be considered routines of the user).

*Results*: Mobility profiles model the systematicity of each user. Fig. 5 (right) depicts the profile extracted in Pisa province, where we can see how some areas become less dense of trips, especially those in the country side and those where systematic trips are less likely to occur (e.g., the seaside, on the left). Fig. 7 (left) shows the number of routines per users in Pisa province, with almost every user having one or two routines, which most likely correspond to commuting trips between home and work. The corresponding average number of routines per profile is 2.14. Fig. 7 (right) reports the temporal distribution of the trajectories and routines. Here we can see that the profiles closely follow the timing of typical working days, highlighting the three peeks during the early morning (5–6), lunchtime (11–12), and late afternoon (17–18).

### 5.2. Carpooling network

In this section we instantiate the network construction step of our proposed methodology, and analyze the characteristics of the resulting carpooling network. In particular, we first focus on the knowledge on mobility that can be inferred from the network, trying to obtain preliminary estimations of the potential reduction of traffic that can result from carpooling initiatives. Then we study the topological properties of the network, computing ranking measures and extracting communities.

#### 5.2.1. Network construction

The carpooling network is derived by the application of the function contained, which defines who can give a lift to whom. Therefore, the resulting network directly depends on the value used for its parameters $spat_{tol}$ and $temp_{tol}$. In order to find good values for these parameters and to obtain a sound network made of reliable carpooling interactions, we performed a network construction test on a sample of 1000 mobility profiles. Fig. 8 shows how the containment is affected, in percentage, in terms of routines and mobility profiles that have at least one match. The default values of $spat_{tol}$ and $temp_{tol}$ are, respectively, 1 km and 30 min. It is worth to notice that by allowing a walking distance ($spat_{tol}$) of 3 km and a wasting time ($temp_{tol}$) of 30 min, about 60% of the profiled users have at least one match, which decreases to 10% if the walking distance becomes 500 m. Similarly, by allowing a walking distance of 1 km and a wasting time of 60 min, 30% of the profiled users have at least one match, which decreases to 10% if the wasting time becomes 15 min. This suggests that an increase in the walking distance has a larger impact than an increase in wasting time, in terms of number of carpooling matches. Based on these observations, we built the carpooling networks for Pisa and Florence using a maximum walking distance of 1 km and a maximum wasting time of 30 min.

#### 5.2.2. Network analysis

*Users classification*: By observing the users appearing in the carpooling networks (among those which have a mobility profile), we can distinguish those that can join others as passengers or drivers, and those that cannot. In

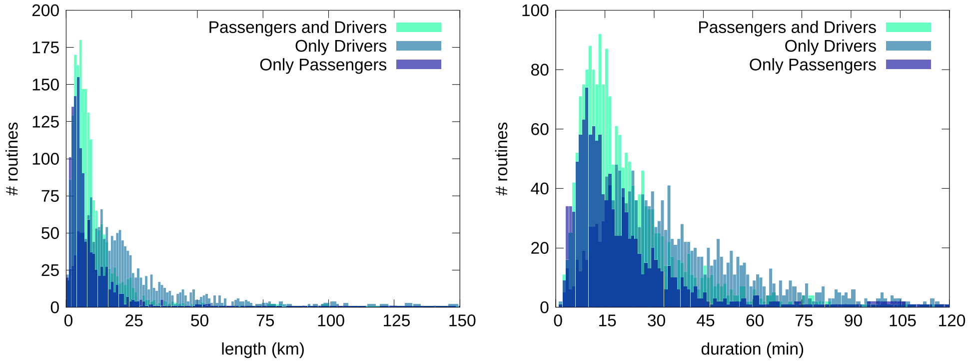



**Fig. 10.** Routines distribution: length (left), duration (center), and time start (right).

particular, we can classify them into four categories, based on their in- and out-degree in the network:

- *only passengers* are the users that can only get rides from other carpoolers, that is $k^{in} = 0$ and $k^{out} > 0$;
- *only drivers* are the users that can only offer rides to other carpoolers, that is $k^{out} = 0$ and $k^{in} > 0$;
- *passengers and drivers* are the users that can act both as passenger and as drivers: $k^{out} > 0$ and $k^{in} > 0$;
- *no carpoolers* are the users that do have systematic movements but cannot share any routines with other users: $k^{out} = 0$ and $k^{in} = 0$.

With respect to the definitions introduced in Section 4.1, users which are only passengers belong to *PP*, those which are only drivers belong to *PD*, and the users which are passengers and drivers belong to both *PP* and *PD*. Fig. 9 depicts the pie chart with the percentages of different types of users in the carpooling user networks of Pisa and Florence. We can observe how the carpooling potentiality is different in the two cities, with Florence showing larger percentages of carpoolers, especially of the *driver and passenger* type.

*Basic carpooling network features*: The analysis of the carpooling network and the corresponding user classification mentioned above can provide a preliminary evaluation of the impact of carpooling services over systematic traffic in terms of reduction of travels and number of cars on the road. For instance, in Pisa we obtained around 7400 mobility routines, each representing at least 8 single trips of the user (indeed, $minsize = 8$ in these experiments), for a total of around 59,200 systematic trips. Also, we discovered that around 1720 of the routines are actually contained in at least one other routine, i.e. the user could carpool with another driver, which means a potential reduction of systematic mobility of about 23%.

We finally analyze the spatio-temporal features of the routines extracted. Fig. 10 (left) shows the length distribution of routines for the categories we described above on the Pisa dataset. We notice that users who are *only passengers* mainly have a routine length between 0 and 10 km, while the *only drivers* have longer routines, between 5 and 25 km. This fact, confirmed by the distribution of trip durations in Fig. 10 (right), meets the intuition that users traveling for longer distances can more easily offer lifts to others, while short-distance travelers can more easily be taken as passengers.

*Topological features*: The following analysis is focused on some topological features of the carpooling user networks, in particular the degree (in-degree $k^{in}$ and out-degree $k^{out}$) of nodes and their ranking scores (driverness $d$ and the passengerness $p$, see Definition 12). The ranking scores are calculated by running the *HITS* algorithm on the carpooling user networks.[3] Fig. 11 shows both the degrees and the ranking scores distribution for Pisa and Florence, with values rescaled to the [0,1] interval in order to make the two plots comparable.

Both distributions are long tailed, meaning that there few users have high values and many users have low values. As highlighted in the previous section, some users are *only passenger* or *only driver*, and therefore their corresponding nodes in the network have $k^{in} = 0$ or $k^{out} = 0$. We can notice that in Pisa, many users also have a zero driverness $d$, and the same happens for passengerness $p$. This emphasizes the significant difference that exists between the degree and the ranking scores, at least in the Pisa carpooling user network. The conclusion is that, despite the obvious correlation between $k^{out}$ and $p$, and between $k^{in}$ and $d$, they can behave in a significant different way, and users that can be drivers for many passengers might possibly be not *good driver*, and vice-versa. On the other hand, the carpooling user network of Florence is denser, and the correlation between degree and ranking scores is higher.

The main differences between the two provinces are in the $p$ and $d$ ranking scores. In Pisa the driverness $d$ rapidly falls down getting close to zero within the first one hundred users, while in Florence it decreases much more gradually. A similar consideration can be done by looking at $p$. Moreover, in Pisa there are few drivers with a high $d$, suggesting that only few of them can serve good passengers, while Florence has more good drivers.

We remark that most of the nodes in the networks considered here have very low degrees, between 3 and 8. This is probably due to the strict parameters that we

[3] For this task we adopted the Python implementation of HITS provided by the NetworkX library (http://networkx.github.io), with a tolerance threshold of 1.0e−8.

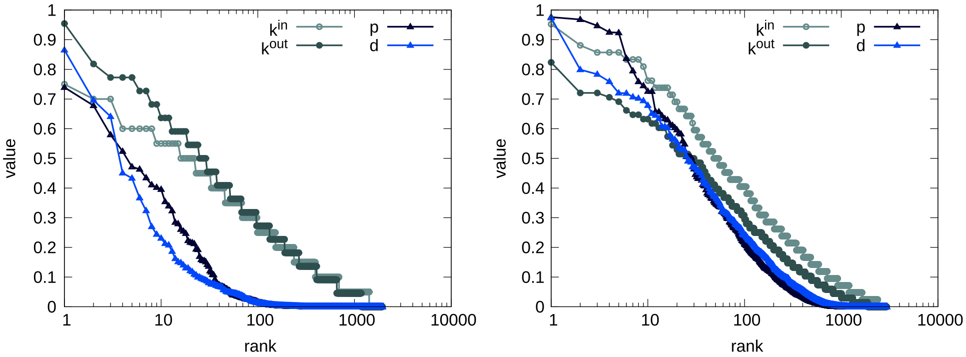



**Fig. 11.** Degree and ranking scores distribution: (left) Pisa and (right) Florence.

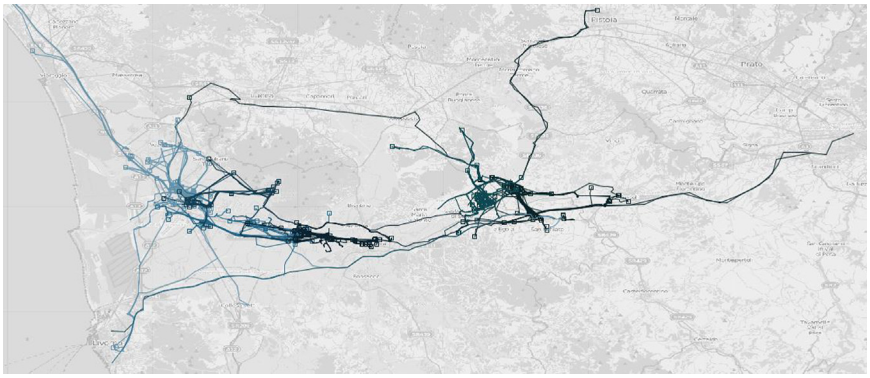


**Fig. 12.** Geographical view of some carpooling communities in Pisa province.

adopted in building the carpooling networks to have reliable interactions. The effect is that the carpooling users networks are very sparse, which turns to be an advantage for the task of suggesting assignments since each user has only a small number choices to consider.

Finally, both datasets show a standard deviation of the all features ($k^{in}$, $k^{out}$, $d$ and $p$) larger than their mean, suggesting that our users are rather heterogeneous. Also, passengerness and driverness appear to be poorly correlated, resulting in a Kendall's Tau coefficient 0.134.

### 5.2.3. *Communities*

*Community discovery*: The HITS algorithm returns an indicator of how much a user can be a good driver or a good passenger. However, these ranking scores do not help in grouping similar users, that is, users that with an high probability would like to share their travels. For this purpose we used carpooling communities, i.e. groups of users who share more routines with other users inside the group than with users outside the group. Various state-of-the-art community discovery algorithms were tested to this purpose, including Infohiermap [37], Louvain [36] and Demon [38]. Finally, the Demon algorithm was selected, due to its better performances both in terms of runtimes and quality of the result. DEMON is an incremental and low-complexity algorithm for community discovery. It is based on the extraction of ego networks, that is, the set of nodes connected with a certain ego node $u$. The communities are extracted by using a bottom-up approach where each node gives the *perspective* of the communities around it, and then all the different perspectives are merged together in an overlapping structure. In practice, the ego network of each node is extracted and the label propagation algorithm is applied on this structure ignoring the presence of the ego itself, since it will be judged by its peer neighbors. Then, with equity, the vote of everyone in the network is combined. The result of this combination is a set of overlapping modules, the guess of the real communities in the global system, made not by an external observer, but by the actors of the network itself. We selected DEMON because the size of the communities are well balanced in opposition to what happens applying the other algorithms (Louvain and Infohiermap). We removed the overlap of the communities by setting to zero the parameter managing the maximum overlap allowed.

*Communities analysis*: Fig. 12 shows a sample of carpooling communities in Pisa province. It is interesting to notice that the carpooling communities are geographically well localized. Every community acts on a specified area that contains the systematic movements of its users. This means, for instance, that a user who is active in the northern area of Pisa can generally disregards the mobility of any user that is moving in the area between Cascina and Pontedera – two cities located 10–30 km East of Pisa.

The topology of the communities emerging from the network results to be very similar to the topology of the original carpooling user network. That is, every community, from a topological point of view, behaves as the

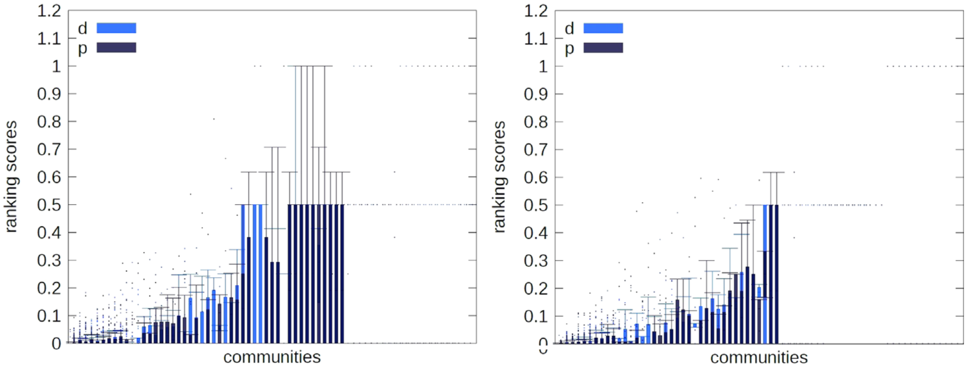



**Fig. 13.** Carpooling ranking scores box-plot for Pisa (left) and Florence (right).

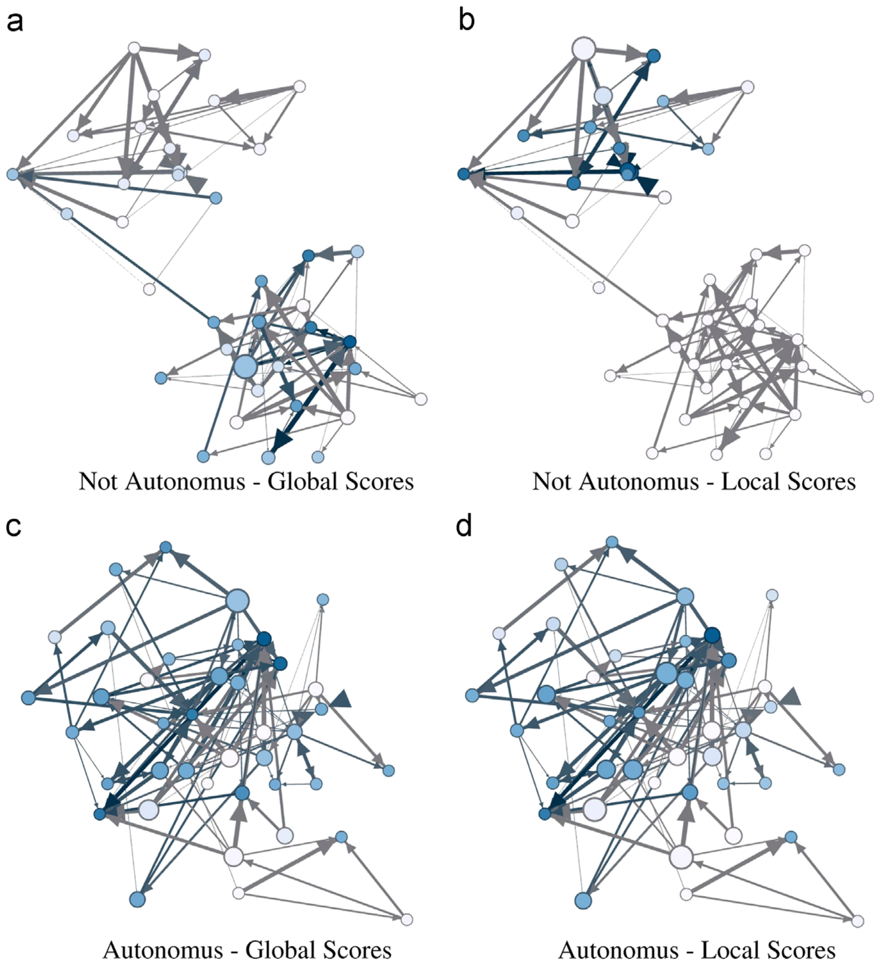



**Fig. 14.** Network of two communities in Pisa, one not autonomous and one autonomous, showing global and local ranking scores. Size of nodes represents driverness. Darkness represents passengerness.

overall network. The average size of the communities is 30–40 nodes and the average degree inside a community is around 4 with a low standard deviation (1.32 on average).

Observing the distribution of the driverness and passengerness scores within each community, shown in Fig. 13 for Pisa and Florence province, we discover that the carpooling communities can be classified in two categories. Indeed, we can see from the box-plots that the distributions on the different communities have a high variability, showing a group of communities having consistently very low values, while the others are made of nodes with (on average) high ranking scores.

As further step, we evaluated how much the ranking scores $d$ and $p$ of a node changes if they computed considering only the community it belongs to, i.e. running the

HITS algorithm locally to the sub-network formed by each community. We call the new scores *local driverness* and a *local passengerness*, to distinguish them from the *global* values. By analyzing the Kendall's tau correlation between the global and local ranking scores for each community we found that, in the Pisa dataset, there are about 30 communities with a correlation close to one, while the remaining circa 20 communities have correlations lower than 0.4. That means that the first group of communities is basically *autonomous*, since they are very weakly influenced by the nodes outside the community, and therefore could rely on finding possible assignments without considering inter-community links. On the contrary, the other communities are *not-autonomous*, since they can be influenced by inter-community links and their users could find potential best matches with users belonging to a different community.

Fig. 14 shows real examples of a *not-autonomous* (left) and a *autonomous* community (right), depicting both the global ranking scores (left column) and the local ones (right column). The size of nodes represents the driverness score, while its darkness represents passengerness. We remark that virtually nothing changes for the *autonomous* community, whereas completely different scores emerge for the *non-autonomous* community, confirming the observations discussed above.

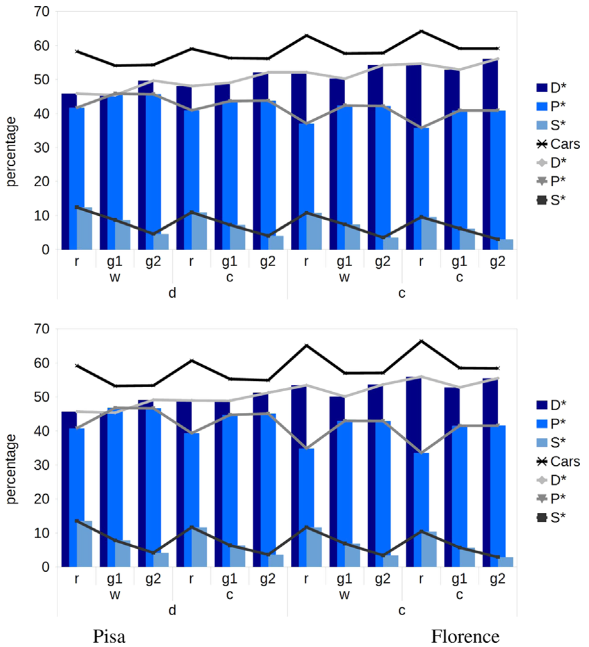



**Fig. 15.** Assignment results for all strategies and criteria adopted: Pisa (left) and Florence (right).

### 5.3. Carpooling suggestions

In this section we describe the results obtained by performing the Never Drive Alone procedure on Pisa and Florence datasets. The assignment performance evaluation is done by measuring the number of resulting SOVs (Single Occupancy Vehicles), the number of systematic cars travelling, as well as evaluating the impact of NDA in economic and environmental terms.

#### 5.3.1. Never drive alone performances

*Setup of experiments*: The NDA procedure described in this paper has been tested considering all the variants discussed in Section 4.5. Moreover, the vehicle capacity of each user has been fixed to $m=4$, i.e. each vehicle can host four passengers in addition to the driver, which fits quite closely the local standards of the area under study. Also, the time slot duration for the creation of temporal networks out of the full carpooling network was fixed to $dur=1$ h, meaning that trips longer than 1 h might be prevented from being matched to others even if the *contain* relation holds – an extremely unlikely event in our dataset, since 1-h routines are very rare.

*Results*: Fig. 15 shows the percentage of passengers $P^*$, drivers with passengers on-board $D^*$ and SOVs $S^*$ obtained over Pisa and Florence by applying each combination of the criteria adopted (abbreviations ($r$), ($g_1$), etc. are those provided in Section 4.5). In addition, it shows the corresponding number of (systematic) cars on the road (see the dark line on the top of both pictures). As first evaluation, we see that there are always more than one-third of users that become passengers, in most cases around half of the users become drivers with passengers, and only a small percentage remains a single-occupant vehicle.

We notice also that while there are significant differences of performances among the algorithm variants considered, the simplest (random) variant already reaches very good results, with a SOV around 12%. Such result suggests that the networks considered constrain significantly the assignment phase, leaving few alternative

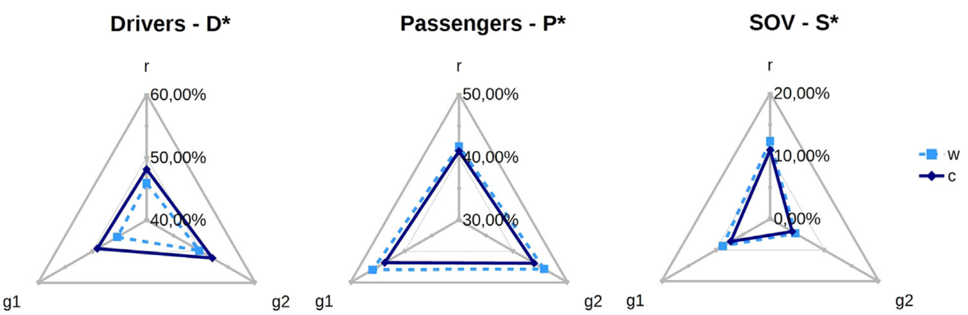



**Fig. 16.** Assignment results for the two edge strategies *(w)* and *(c)* and for the three sorting criteria adopted on the Pisa dataset.

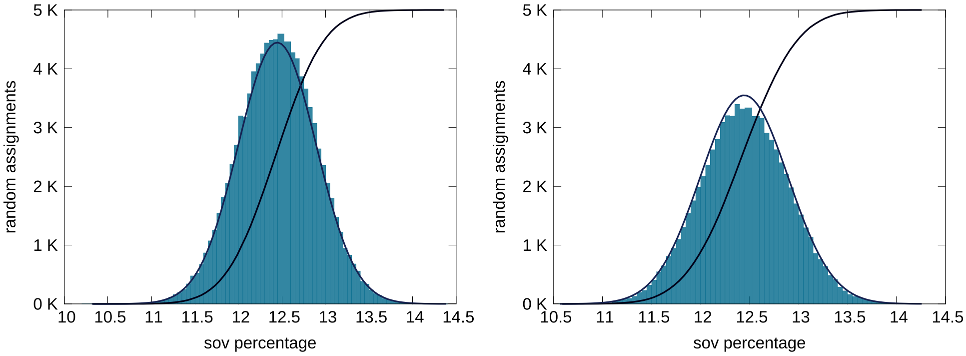



**Fig. 17.** SOVs percentage distribution (PDF and CDF) of random assignment tests ran 100,000 times for discrete time strategies. Left: case not considering communities; right: communities are considered.

opportunities to explore, although smarter assignment methods are able to improve the results. More tolerant settings in the construction of the carpooling network (such as admitting matched with longer distances to walk to take a lift) are expected to yield networks with more alternatives to explore, and therefore make the improvement margins over the random solution much larger.

The plots show that the knowledge extracted from the mobility data and refined with network analysis progressively leads to improvements regarding the minimization of the number of SOVs. Indeed, we observe that the sorting criterion $(g_2)$ gets better results than the sorting criterion $(g_1)$, which in turn outperforms $(r)$.

Moreover, Fig. 16 also depicts how the strategy considering the community information $(c)$ slightly reduces the number of SOVs with respect to the strategy that considers the whole network $(w)$. This suggests that the carpooling service might be organized in a local way, i.e. it might be convenient to focus the proactive suggestions mainly among users within the same community, basically disregarding the others.

Also the temporal information contributes with useful suggestions: considering dynamically each change in the carpooling interactions $(d)$ to compute the assignments procures a little advantage with respect to the one obtained using fixed time slots $(s)$. Yet, the calculus with $(d)$ is computationally more expensive, especially in periods where carpooling interactions are frequent (morning, midday, evening).

So far, our considerations were focused on the minimization of the number of SOVs. Anyway, if we want primarily to minimize the number of systematic cars traveling, and only secondarily the number of SOVs, we discover that the best approach still uses the $(g_2)$ criteria, yet this time considering the whole network $(w)$ and static (discrete) time slots.

Finally, Fig. 15 also shows that although Florence has more good drivers and passengers than Pisa (see the carpooling user network analysis in Section 5.2), the two carpooling networks yield comparable results in terms of suggestions.

*NDA vs. random assignment approach*: In order to better verify that the provided solution is consistently better than those found by a random exploration of choices, we report in Fig. 17 the results obtained by running 100,000 times NDA with random sorting criteria $(r)$ on the Pisa carpooling network, considering the whole network without assignment priorities (left $(w)$) and prioritizing the assignments between nodes in the same community (right $(c)$). What we obtain in both cases is a normal distribution. Regarding $(w)$ the mean value of SOVs, obtained nearly five thousand times, is 12.44 and the standard deviation is 1.48. On the other hand, considering $(c)$, the mean value is 12.28, a bit lower than the previous, but obtained no more than three thousand times and a half, and with a larger standard deviation of 1.97. The solution provided by NDA considering both carpooling ranking measures and community knowledge provides a SOVs percentage slightly smaller than 4.63%, which is largely better than anyone found by the 100,000 random runs. Indeed, according to the distributions shown in the figure the expected probability of finding a SOVs percentage lower than that is around $6.56 \times 10^{-8}$, therefore very close to zero.

*Comparison with existing approaches*: As described in the related works, most of the literature on carpooling is focused either on the simulation of very specific aspects (such as the impact of high occupancy vehicle lanes on traffic) or on the realization of a real-time service. On the opposite, our work aims to provide a solution for carpooling matching and study its impact in a real context. The main works that tackle problems close to ours are [29,22], which we considered for a comparison of performances. Both works are based on data sources significantly different from those adopted in our paper: [29] is tailored around (geo-localized) Twitter data, and exploits the topics of the text messages posted and the social network of users; [22], instead, is based on a mix of mobile phone data (CDR traces) and social media (geo-localized Twitter posts and Foursquare check-ins). That makes a direct (and fair) comparison over a common benchmark very difficult.

Another important difference between our approach and the two competitors considered is that the latter aim to maximize the number of users involved in the carpooling, yet not considering explicitly the overall coherence of the carpooling assignment, i.e. a passenger for a home-to-work trip needs to be passenger also for the return trip. In the following summary of results, we call this incomplete form of assignment *partial passengers*, in contrast to the complete one, called *total passengers*. As described in Section 4, our approach is focused on the

Please cite this article as: R. Guidotti, et al., Never drive alone: Boosting carpooling with network analysis, Information Systems (2016), http://dx.doi.org/10.1016/j.is.2016.03.006

more realistic scenario of *total passengers*, which is ensured by requiring that the status of the user (passenger or driver) is kept for the whole day.

Below we provide an indirect comparison of the three methods, summarizing the performance results obtained by each of them over its own datasets:

- *CAR-O* [29]: 71.95% of users in the Rome dataset and 74.82% of users in San Francisco become *partial passengers*. Impact on single trips saved not provided.
- *EN-ROUTE* [22]: 65% of users in Madrid and 68% in New York become *partial passengers*. Impact on single trips saved not provided.
- *NDA*: 43.83% of users in Pisa and 45.10% in Florence become *total passengers*. Impact on single trips is 77.52% in Pisa, and 77.03% in Florence.

These results suggest that the matching strategies provided by our solution can reach an impact over car traffic that is apparently similar to those obtained by other approaches in similar contexts, yet providing a more realistic application scenario.

**Table 1**
Number of routines extracted in the two cities, the routines that are linked to others in the carpooling network, those that might be served by others, those that might serve at least another one, and number of matches found by Never Drive Alone (also in percentage w.r.t. potential passengers).

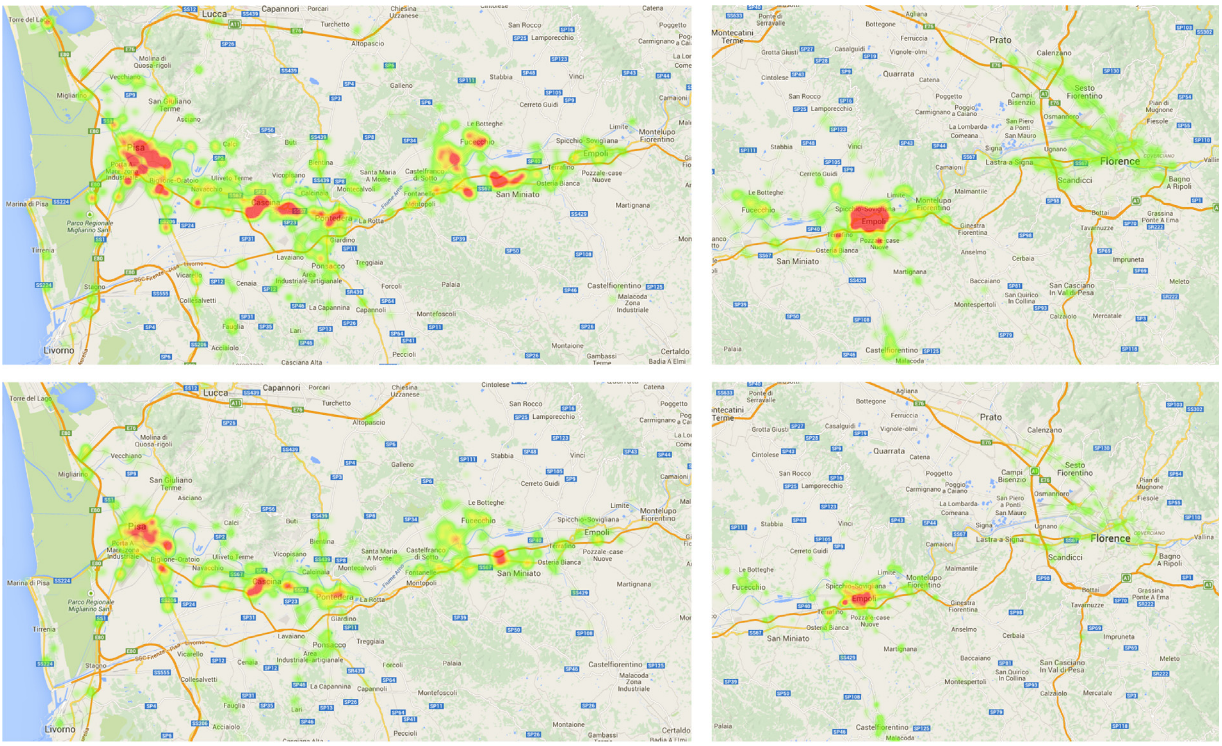
| City | # routines | # linked | # can ride | # can drive | # saved trips (%) |
|---|---|---|---|---|---|
| Pisa | 7383 | 3049 | 1717 | 1995 | 1331 (77.52) |
| Florence | 9801 | 5712 | 3305 | 4140 | 2546 (77.03) |

**Table 2**
Estimates of total potential savings in a normal day obtained by using the proactive carpooling proposed in this work. Savings are expressed in terms of total kilometers driven, time spent driving, fuel consumed, its cost and $CO_2$ emissions.

| City | km | min | *Fuel* (l) | € | $CO_2$ (kg) |
|---|---|---|---|---|---|
| Pisa | 10,868.36 | 24,174.58 | 646.67 | 1001.49 | 1445.49 |
| Florence | 16,748.99 | 43,300.28 | 996.56 | 1543.37 | 2227.62 |

### 5.3.2. *Evaluating the economic and environmental impact of carpooling*

In order to evaluate the practical importance of the carpooling matching discussed in the previous section, we consider here the best configuration setting for the system and study its results from several viewpoints. The first one is simply the impact of the carpooling in terms of reduction of cars on road. Table 1 summarizes the number of routines observed in the two showcases with details on the number of routines that might potentially be served by other drivers (# *can ride*), those that might give a lift to other passengers (# *can drive*) and their union (# *linked*). Finally, the number of matches that were actually found by the algorithm is also in terms of percentage over the



Pisa Florence

**Fig. 18.** Spatial distribution of pick-up and drop-off points of NDA solution. *Left column*: Pisa province; *right column*: Florence province. *First row*: pick-up points; *second row*: drop-off points.

**Table 3**
Runtimes of the different phases of NDA over the Pisa and Florence datasets. Notice that a single value is provided for the Profiles construction, since it is basically independent from the specific area/user network the users belong to.

| City | # users | Profiles (s/user) | Carpooling network (s) | User net. (s) | Temporal net. (s) | HITS (s) | Assign (s) |
|---|---|---|---|---|---|---|---|
| Pisa | 2168 | 62,5 | 2460 | $<1$ | $<1$ | 21.8 | 2 |
| Florence | 3324 | | 3120 | $<1$ | $<1$ | 23.8 | 3 |

maximum theoretical outcome, i.e. the number of potential passengers. As we can see, NDA is able to assign most part of the potential passengers in both cities (around 77% of them), also corresponding to a relevant percentage of total routines (cars on road) saved, namely 18% in Pisa and 26% in Florence.

Tables 2 reports the economic and environmental impact that the traffic reductions obtained with carpooling can have. Estimates of such impact are computed considering the most common car sold in the period of data collection, an average gasoline consumption of 0.0595 l/km, a gasoline cost in the observation period of 1.54869 € per liter, and a $CO_2$ emission of 133 g per km.[4] Considering that the estimates reported in the table are relative to a single city and a single (typical) day, the reduction values are very significant, especially towards the environment.

Finally, we show in Fig. 18 the spatial distribution of pick-up (top row) and drop-off (bottom row) points of the solution found by NDA on Pisa (left) and Florence (right). We can see that in the case of Pisa, carpooling mainly (yet not exclusively) involves several smaller cities distributed along an important road towards East, connecting Pisa with the other major cities of the region. For Florence it is interesting to notice that a major hotspot, even larger than Florence itself, is located in a nearby city, Empoli, characterized by a huge flow of commuters towards Florence an the surrounding industrial areas. In general, carpooling is much more concentrated around a few dense areas than what happens for Pisa. In both cases the drop-off points appear to be more concentrated around the main attractors, while pick-up points are slightly more dispersed.

### 5.4. Runtimes

An empirical evaluation of the runtimes of each step of the proposed solution has been performed, and summarized in Table 3 for the two provinces considered.

The results say that the most expensive operations are the extraction of mobility profiles and the construction of the carpooling network, i.e. the preprocessing tasks that precede the actual computation of scores and the assignment. The extraction of profiles is local to the single user, and therefore independent from the network size and structure of the dataset considered. In the present implementation it takes around 1 min of computation for each user. Building the carpooling network has a theoretical quadratic cost in the number of routines, yet, the software developed includes spatial filters that quickly recognizes clearly incompatible profiles, and which are more effective in larger areas, which is the case of Florence. For this reason, the actual runtime for computing the network grew much more slowly when moving from the smaller dataset (Pisa) to the larger one (Florence). The other steps, including the final assignment algorithm, have much smaller costs that do not affect the overall times significantly.

### 5.5. Summary of results

The various experiments presented in this section describe the several faces of the approach we propose. In the following we provide a short recap of the key results obtained and lessons learned.

Analyzing the routines of users, typically two symmetrical routines – home-to-work and work-to-home – emerge, i.e. the routines follow the timing of typical working days and summarize the overall collective movements.

We discovered that indicators derived from the carpooling network, like number of only drivers/only passengers/passengers and drivers, can be used to characterize different areas and cities in terms of applicability of carpooling. Also, a measure of empirical upper bound of the potential reduction of cars on the road can be inferred, whose average in the area of our experimentation is around 23%.

The carpooling networks tend to be very sparse, and are characterized by long tailed distributions both for the in–out-degree and for the driverness and passengerness indexes. Also, carpooling networks can usually be partitioned into sub-networks which are well isolated from other components and can be treated separately, each having pools of potential passengers and drivers.

The heuristics for carpooling assignments we developed greatly benefits from the knowledge provided by the driverness and passengerness scores, as well as the fragmentation into communities. Performances show a percentage of single occupancy vehicles (SOVs) as low as 4.63%, which is less than half of what any random assignment can reach in practice.

As overall result, about 77% of the trips could be saved on both datasets, and the estimates of saved kms, time, fuel, money and $CO_2$ emissions are significant.

## 6. Conclusion and future work

In this paper we have proposed a novel approach for analyzing the potentiality of a carpooling service and for

[4] http://www.patentati.it/blog/articoli-auto/classifica-auto-2011.html, http://dgerm.sviluppoeconomico.gov.it/dgerm/prezzimedi.asp?anno=2011, http://www.ilsole24ore.com/speciali/emissioni.

suggesting an assignment among systematic car drivers in order to have them not to drive alone. Many useful observations for a carpooling service resulted from our study. We showed how ranking measures and communities extracted from mobility networks can be used to characterize different aspects of human mobility. By exploiting them, we proposed an approach for boosting carpooling using network analysis. Moreover, we have seen that the ranking values distributions characterize in a different way for different geographical areas. Furthermore, we have found that carpooling communities can be classified into two categories: autonomous communities, that, being independent from the rest of the car drivers, are made by many good carpoolers offering and taking lifts to many users; non-autonomous communities, that being influenced by extra community car drivers, cannot be managed on their own. A suggestion from this last point is that if a new carpooling service is to be realized, a good start point would be autonomous communities. Finally, we saw how the potential carpooling network can be used to suggest assignments among systematic car drivers and how ranking measures considered on communities lead to valuable reductions of the cars employed in systematic mobility. In particular, we have shown how the conjunctive application of these features lead to valuable performances in terms of assignments and reduction of SOVs.

Our task is obviously a starting point with respect to the proposal of a real carpooling service. For example, it could be considered in the matching phase that a passenger is willing to wait or to walk a bit more for a long travel then for a short one. Moreover, instead of considering matches only between systematic movements, it could be interesting to consider the number of non-systematic movements that can be saved. Thanks to the proposed approach, the knowledge about systematic behavior, and the measures regarding carpooling, could really help our everyday life in reducing traffic, saving money and producing less pollution.

## Acknowledgments

This work has been partially supported by the European Commission under the FET-Open Project no. FP7-ICT-284715, ICON, and by the European Commission under the SMARTCITIES Project no. FP7-ICT-609042, PETRA.

## References

[1] R. Trasarti, F. Pinelli, M. Nanni, F. Giannotti, Mining mobility user profiles for car pooling, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'11, ACM, New York, NY, USA, 2011, pp. 1190–1198.
[2] J. Dahlgren, High occupancy vehicle lanes: Not always more effective than general purpose lanes, Transport. Res. Part A: Policy Pract. 32 (2) (1998) 99–114.
[3] H. Yang, H.-J. Huang, Carpooling and congestion pricing in a multilane highway with high-occupancy-vehicle lanes, Transport. Res. Part A: Policy Pract. 33 (2) (1999) 139–155.
[4] J. Kwon, P. Varaiya, Effectiveness of California's high occupancy vehicle (hov) system, Transport. Res. Part C: Emerg. Technol. 16 (1) (2008) 98–115.
[5] G. Giuliano, D.W. Levine, R.F. Teal, Impact of high occupancy vehicle lanes on carpooling behavior, Transportation 17 (2) (1990) 159–177.
[6] H. Konishi, S.-i. Mun, Carpooling and congestion pricing: Hov and hot lanes, Region. Sci. Urban Econ. 40 (4) (2010) 173–186.
[7] A.M. Bento, J.E. Hughes, D. Kaffine, Carpooling and driver responses to fuel price changes: evidence from traffic flows in Los Angeles, J. Urban Econ. 77 (2013) 41–56.
[8] M. Sghaier, H. Zgaya, S. Hammadi, C. Tahon, A distributed Dijkstra's algorithm for the implementation of a real time carpooling service with an optimized aspect on siblings, in: 2010 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, Madeira Island, Portugal, 2010, pp. 795–800.
[9] M. Armendáriz, J. Burguillo, A. Peleteiro, G. Arnould, D. Khadraoui, Carpooling: a multi-agent simulation in netlogo, in: Proceedings of the ECMS, 2011, 61–67.
[10] T. Bellemans, S. Bothe, S. Cho, F. Giannotti, D. Janssens, L. Knapen, C. Körner, M. May, M. Nanni, D. Pedreschi, et al., An agent-based model to evaluate carpooling at large manufacturing plants, Proc. Comput. Sci. 10 (2012) 1221–1227.
[11] S. Cho, A.-U.-H. Yasar, L. Knapen, T. Bellemans, D. Janssens, G. Wets, A conceptual design of an agent-based interaction model for the carpooling application, Proc. Comput. Sci. 10 (2012) 801–807.
[12] S. Galland, L. Knapen, A.-U.-H. Yasar, N. Gaud, D. Janssens, O. Lamotte, A. Koukam, G. Wets, Multi-agent simulation of individual mobility behavior in carpooling, Transport. Res. Part C: Emerg. Technol. 45 (2014) 83–98.
[13] S. Galland, N. Gaud, A.-U.-H. Yasar, L. Knapen, D. Janssens, O. Lamotte, Simulation model of carpooling with the janus multiagent platform, Proc. Comput. Sci. 19 (2013) 860–866.
[14] S. Galland, A. Yasar, L. Knapen, N. Gaud, D. Janssens, O. Lamotte, G. Wets, A. Koukam, Multi-agent simulation of individual mobility behavior in carpooling using the janus and jasim platforms, Transport. Res. Part C.
[15] R.F. Teal, Carpooling: who, how and why, Transport. Res. Part A: General 21 (3) (1987) 203–214.
[16] R. Trasarti, F. Pinelli, M. Nanni, F. Giannotti, Mining mobility user profiles for car pooling, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 1190–1198.
[17] G. Correia, J.M. Viegas, Carpooling and carpool clubs: Clarifying concepts and assessing value enhancement possibilities through a stated preference web survey in Lisbon, Portugal, Transport. Res. Part A: Policy Pract. 45 (2) (2011) 81–90.
[18] T. Vanoutrive, L. Van Malderen, B. Jourquin, I. Thomas, A. Verhetsel, F. Witlox, Carpooling and employers: a multilevel modelling approach, in: 3rd Transport Research Day, VUB Press, 2009, pp. 335–349.
[19] T. Vanoutrive, E. Van De Vijver, L. Van Malderen, B. Jourquin, I. Thomas, A. Verhetsel, F. Witlox, What determines carpooling to workplaces in Belgium: location, organisation, or promotion? J. Transport Geogr. 22 (2012) 77–86.
[20] D. Lee, S.H. Liang, Crowd-sourced carpool recommendation based on simple and efficient trajectory grouping, in: Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science, ACM, 2011, pp. 12–17.
[21] K.W. Steininger, G. Bachner, Extending car-sharing to serve commuters: an implementation in Austria, Ecol. Econ. 101 (2014) 64–66.
[22] B. Cici, A. Markopoulou, E. Frias-Martinez, N. Laoutaris, Assessing the potential of ride-sharing using mobile and social data: a tale of four cities, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2014, pp. 201–211.
[23] N. Bicocchi, M. Mamei, Investigating ride sharing opportunities through mobility data analysis, Pervasive and Mobile Computing.
[24] D.W. Massaro, B. Chaney, S. Bigler, J. Lancaster, S. Iyer, M. Gawade, M. Eccleston, E. Gurrola, A. Lopez, Carpoolnow-just-in-time carpooling without elaborate preplanning, in: WEBIST, 2009, pp. 219–224.
[25] L. Knapen, D. Keren, A.-U.-H. Yasar, S. Cho, T. Bellemans, D. Janssens, G. Wets, Estimating scalability issues while finding an optimal assignment for carpooling, Proc. Comput. Sci. 19 (2013) 372–379.
[26] D. Coppersmith, T. Nowicki, G. Paleologo, C. Tresser, C.W. Wu, The optimality of the online greedy algorithm in carpool and chairman assignment problems, ACM Trans. Algorithms (TALG) 7 (3) (2011) 37.
[27] R. Tijdeman, The chairman assignment problem, Discrete Math. 32 (3) (1980) 323–330.

Please cite this article as: R. Guidotti, et al., Never drive alone: Boosting carpooling with network analysis, Information Systems (2016), http://dx.doi.org/10.1016/j.is.2016.03.006

[28] V. Lerenc, Increasing Throughput for Carpool Assignment Matching, US Patent App. 13/329,899, December 19, 2011.
[29] R. Guidotti, A. Sassi, M. Berlingerio, A. Pascale, B. Ghaddar, Social or green? a data-driven approach for more enjoyable carpooling, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2015, pp. 842–847.
[30] P. Minett, J. Pearce, Estimating the energy consumption impact of casual carpooling, Energies 4 (1) (2011) 126–139.
[31] D. Zhang, Y. Li, F. Zhang, M. Lu, Y. Liu, T. He, Coride: carpool service with a win–win fare model for large-scale taxicab networks, in: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, ACM, 2013, p. 9.
[32] M. Van Vugt, P.A. Van Lange, R.M. Meertens, J.A. Joireman, How a structural solution to a real-world social dilemma failed: a field experiment on the first carpool lane in Europe, Soc. Psychol. Q. 59 (1996) 364–374.
[33] K.K. Charles, P. Kline, Relational costs and the production of social capital: evidence from carpooling*, Econ. J. 116 (511) (2006) 581–604.
[34] D. Easley, J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.
[35] M. Coscia, F. Giannotti, D. Pedreschi, A classification for community discovery methods in complex networks, Stat. Anal. Data Min. 4 (5) (2011) 512–546.
[36] V.D. Blondel, J. Loup Guillaume, R. Lambiotte, E. Lefebvre, Fast Unfolding of Communities in Large Networks (2008).
[37] M. Rosvall, C.T. Bergstrom, Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated Systems, CoRR abs/1010.0431.
[38] M. Coscia, G. Rossetti, F. Giannotti, D. Pedreschi, Demon: a local-first discovery method for overlapping communities, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'12, ACM, New York, NY, USA, 2012, pp. 615–623.
[39] M.R. Korupolu, C.G. Plaxton, R. Rajaraman, Analysis of a local search heuristic for facility location problems, J. Algorithms 37 (1) (2000) 146–188.
[40] http://www.octotelematics.com/it (2014).
[41] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, Optics: ordering points to identify the clustering structure, in: ACM SIGMOD Record, ACM, 1999, pp. 49–60.

Please cite this article as: R. Guidotti, et al., Never drive alone: Boosting carpooling with network analysis, Information Systems (2016), http://dx.doi.org/10.1016/j.is.2016.03.006