

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, Dep. of Computer Science, University of Pisa, Largo B. Pontecorvo, 3, Pisa, Italy*

^b*KDDLab, ISTI-CNR, Via G. Moruzzi, 1, Pisa, Italy*

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.

Keywords: Carpooling; Complex Networks; Rank Analysis; Community Discovery; Big Data Analytics; Mobility Data Mining

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

Email 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)

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 digitised trajectories of vehicular travels recorded by GPS-enabled on-board devices. These forms of *big data* has 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 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 analysing 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 sub-populations 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 analyse the potential aggregated outcome of carpooling in the analysed 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 behaviour.

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 from 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 literature.

Carpooling received wide attention in the theoretical literature, mainly regarding high occupancy vehicle lanes (HOV) [2, 3, 4, 5, 6, 7]. [2, 4] develops models to calculate the benefits gained for eliminating traffic congestion by adding HOV lanes, or by converting general purpose lanes into HOV. [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] and [6], consider tolls related with HOV and how these can influence their use. [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, 9, 10, 11, 12, 13]. A multi-ABM in conjunction with the Dijkstra's algorithm is used in [8] to efficiently answer real time users' queries. In [14] 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. [11, 15, 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 [16, 17, 18, 19, 20, 21, 22, 23]. In [16], 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 [17], it is introduced a methodology for extracting mobility profiles of individuals, and a study criteria to match common routes in order to develop a carpooling service. Something similar is illustrated in [23], 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. [18] 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 carpools employees at a workplace, [19, 20] predict the share of carpools at large workplaces locations, organization and carpools promotion. In [22] 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 [24] the authors develop an application for car sharing recommendation by exploiting a topic clustering algorithm applied to labeled trajectories.

In other studies [25, 26, 27, 21], 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. [25] 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 [27], the problem is faced as an optimization task reduced to the *chairman assignment problem* [28]. [29] considers simulated straight-line trajectories observing only origin and destination of trips and classifies users as eligible or ineligible for carpools by minimizing the time of the trip. In [26] a user network is built that represents planned periodic trips, where the edges are labelled with the probability of negotiation success for carpools. 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 [26] and ours is that we provide matches between couples of users in a pro-active way, suggested from data and not advertised from people. Moreover, [26] 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. [21] develops a methodology that finds feature points in trajectories and organize them in a trie data structure to speed up and refine geographical queries for carpools purposes.

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 carpools. In [31] instead, the authors propose a carpooling based on taxicab, that is, they analyze the reduction of circulating taxi in 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 through-

out 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 Blablacar¹ 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 focus on commuters. The most popular services in the area where our case studies are located (Italy) include Autostradecarpooling, Avacar, Bring-me, Viaggiainsieme, and Autoincomune². 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 neighbouring 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:

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

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 $H_u = \{T_1, \dots, T_k\}$.

Using the above definitions and following the profiling procedure proposed in [17], we can retrieve the systematic movements of a certain user u . 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 behaviour:

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/>

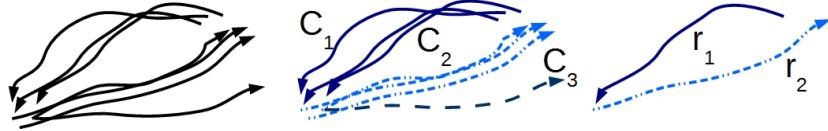


Figure 1: The user’s *individual history* (left: black lines), the clusters identified by the grouping function (center: C_1, C_2, C_3) and the extracted *individual routines* (right: r_1, r_2) forming her *individual mobility profile*.

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

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 ϵ a distance threshold. Given a partitioning function $\text{Profile}(H_u, ms, \text{Dist}, \epsilon) = \mathcal{M} = \{\mathcal{M}_1, \dots, \mathcal{M}_k\}$, with $H_u \subseteq \bigcup_{i=1}^k \mathcal{M}_i$ and $\forall 1 \leq i < j \leq k. \mathcal{M}_i \cap \mathcal{M}_j = \emptyset$, for each $1 \leq i \leq k$ 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 $P_u = \{r_1 \dots r_k\}$. The residual trajectories, i.e. $H_u \setminus \bigcup_{i=1}^k \mathcal{M}_i$, represent occasional trips and do not contribute to any routine in the user mobility profile.

Following [17], 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 $\text{ranking}(G)$ 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 literature, several popular community discovery algorithms exist [36, 37, 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 $\text{communities}(G)$ as the algorithm Infohiermap, taking as input G and returning a set of communities $\mathcal{C} = \{C_1 \dots C_n\}$, 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 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 $r_1 = \{(x_1^{(1)}, y_1^{(1)}, t_1^{(1)}), \dots, (x_n^{(1)}, y_n^{(1)}, t_n^{(1)})\}$ and $r_2 = \{(x_1^{(2)}, y_1^{(2)}, t_1^{(2)}), \dots, (x_m^{(2)}, y_m^{(2)}, t_m^{(2)})\}$, a spatial tolerance $spat_{tol}$ and a temporal tolerance $temp_{tol}$, we say that r_1 is contained in r_2 , i.e. $contained(r_1, r_2, spat_{tol}, temp_{tol})$, if $\exists i, j, 1 \leq i < j \leq m$ such that:

$$\begin{aligned} &||(x_1^{(1)}, y_1^{(1)}) - (x_i^{(2)}, y_i^{(2)})|| + ||(x_n^{(1)}, y_n^{(1)}) - (x_j^{(2)}, y_j^{(2)})|| \leq spat_{tol} \wedge \\ &|t_1^{(1)} - t_i^{(2)}| + |t_n^{(1)} - t_j^{(2)}| \leq temp_{tol} \end{aligned}$$

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.

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 $\mathcal{P} = \{P_1, \dots, P_n\}$, 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 $\{(u, v, r_i^u, r_j^v)\} \in E$.

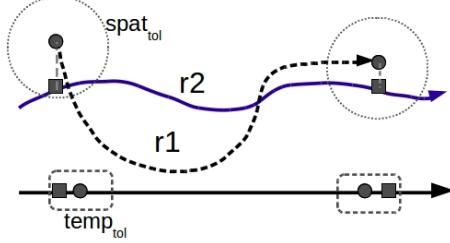


Figure 2: Example of routines containment: r_1 is contained in r_2 because the starting and ending points of r_1 (circular points) are spatially and temporally close enough to some points of r_2 (squared points).

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 Figure 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.

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 Figure 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. $E' = \{(u, v) | (u, v, r_i^u, r_j^v) \in E\}$.

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 .

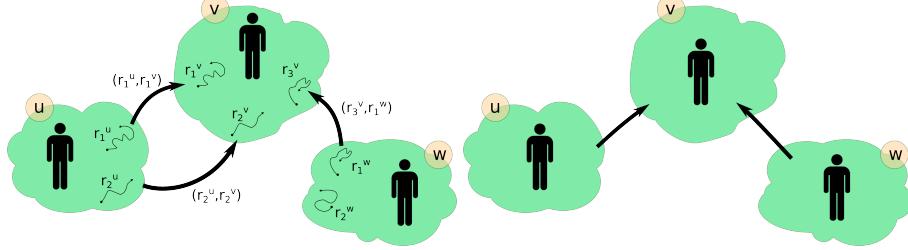


Figure 3: Carpooling Network (left), Carpooling User Network (right).

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. multi-dimensional edges are not considered, assuming that each pair of users can share only one routine: the general case will be described later as 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 $|N| = |D| + |P| + |SOV|$.

The procedure uses a sorting function f to order the *possible passengers* according to some criteria c' . It takes the first possible passenger u from the sorted list, and it orders her *possible drivers* (i.e. the out-linked nodes in the network) using f according to another criteria c'' . Then, it takes the first driver v that still has free places in her car, and assigns u to v . The procedure is repeated until every user is assigned, or there are no free places left. The *greedy assignment* method is illustrated in Algorithm 1 where the function $\text{successors}(u)$ returns the set of successors of u .

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 $\Theta(N \log N)$.

The problem analyzed is NP-complete [39], and an optimal approach to solve it is exponential in the number of edges. Indeed, the approach followed to get an optimal solution must take into account the fact that every assignment might inhibit any of the others, 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.

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 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.

ALGORITHM 1: *calculateGeedyAssignment*($G', f, m, c', c'', D, P, S$)

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 , $|P_u| \leq 1$ always) S - set of single occupant vehicle

Output: D, P, S

```

for  $u \in f(N, c')$  do
    if  $D_u \not\subseteq D \wedge P_u \not\subseteq P$  then
        for  $v \in f(\text{successors}(u), c'')$  do
            if  $|D_v| \leq m$  then
                 $D_v \leftarrow D_v \cup \{u\};$ 
                 $P_u \leftarrow \{v\};$ 
                break;
            end
        end
    end
end
for  $u \in N$  do
    if  $D_u \not\subseteq D \wedge P_u \not\subseteq P$  then
         $S \leftarrow S \cup \{u\};$ 
    end
end
return  $D, P, S;$ 

```

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 have more interest to travel together, or which geographical areas could be more suitable for the carpooling service. Consequently, we extracted groups of users sharing common routines to discover how they are characterized from a geographical point of view, 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:

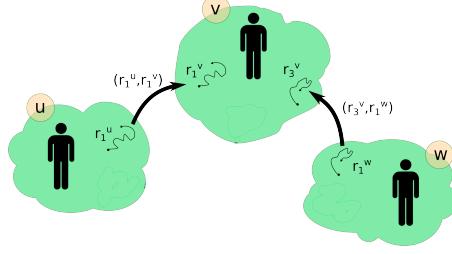


Figure 4: Carpooling Temporal Network.

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 $E' = \{e_{uv} \in E \mid \text{isActive}(e_{uv}, ts, dur)\}$ and $N' \subseteq N$ is the set of all nodes comparing in E' . The isActive operator is defined as

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

Where $t_1^{r_i}$ is the time stamp of the first point of r_i and $t_n^{r_i}$ is the time stamp of the last point of r_i .

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 phenomena 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.

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. NDA performs the following steps: (i) extracts the systematic movements; (ii) builds the carpooling network; (iii) calculates the *passengeress* and *driverness* ranking scores; (iv) extracts the carpooling communities; (v) makes the assignments and classify the users as drivers, passengers or 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.

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 than m (max number of free places) users because if she systematically travels a long routine, she might drop-off

ALGORITHM 2: *NeverDriveAlone(\mathcal{M}, dur, f, m)*

Input : \mathcal{M} - 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
 $\text{for } M_u \in \mathcal{M} \text{ do}$ 
     $| Pr_u \leftarrow \text{Profile}(M_u);$ 
     $| \mathcal{P} \leftarrow \mathcal{P} \cup Pr_u;$ 
 $\text{end}$ 
 $G \leftarrow \text{buildCarpoolingNetwork}(\mathcal{P}, \text{contained}(*));$ 
 $G' \leftarrow \text{extractCarpoolingUserNetwork}(G);$ 
 $k^{out}, k^{in} \leftarrow \text{getDegrees}(G'); /*$  calculates out-degree and in-degree values
 $p, d \leftarrow \text{HITS}(G'); /*$  calculates passangerness and driverness ranking scores
 $c' \leftarrow \text{createSortingCriteria}(k^{out}, p); /*$  creates the first sorting criteria
 $c'' \leftarrow \text{createSortingCriteria}(k^{in}, d); /*$  creates the second sorting criteria
 $D' \leftarrow \emptyset; P' \leftarrow \emptyset; S' \leftarrow \emptyset;$ 
 $\text{for selected } ts \text{ do}$ 
     $| G^{ts, ts+dur} \leftarrow \text{extractCarpoolingTemporalNetwork}(G, ts, dur);$ 
     $| D', P', S' \leftarrow \text{calculateGeedyAssignment}(G^{ts, ts+dur}, f, m, c', c'', D', P', S');$ 
     $| D, P, S \leftarrow \text{updateAssignments}(D, P, S, D', P', D');$ 
     $| D', P', S' \leftarrow \text{removeFinishedInteractions}(G^{ts, ts+dur}, D', P', S', ts, dur);$ 
 $\text{end}$ 
return  $D, P, S;$ 

```

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 3: *NeverDriveAloneCommunities(\mathcal{M}, dur, f, m)*

Input : \mathcal{M} - 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
 $\text{for } M_u \in \mathcal{M} \text{ do}$  *
     $| Pr_u \leftarrow \text{Profile}(M_u);$ 
     $| \mathcal{P} \leftarrow \mathcal{P} \cup Pr_u;$ 
 $\text{end}$ 
 $G \leftarrow \text{buildCarpoolingNetwork}(\mathcal{P}, \text{contained}(*));$ 
 $G' \leftarrow \text{extractCarpoolingUserNetwork}(G);$ 
 $k^{out}, k^{in} \leftarrow \text{getDegrees}(G'); /*$  calculates out-degree and in-degree values */
 $p, d \leftarrow \text{HITS}(G'); /*$  calculates passangerness and driverness ranking scores */
 $\mathcal{C} \leftarrow \text{extractCommunities}(G'); /*$  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 */
 $D' \leftarrow \emptyset; P' \leftarrow \emptyset; S' \leftarrow \emptyset;$ 
 $\text{for selected } ts \text{ do}$ 
     $| G^{ts, ts+dur} \leftarrow \text{extractCarpoolingTemporalNetwork}(G, ts, dur);$ 
     $| \text{for } C \in \mathcal{C} \text{ do}$ 
         $| | G_C^{ts, ts+dur} \leftarrow \text{extractSubGraph}(G^{ts, ts+dur}, C);$ 
         $| | D', P', S' \leftarrow$ 
         $| | \text{calculateGreedyAssignment}(G_C^{ts, ts+dur}, f, m, c', c'', D', P', S');$ 
     $| \text{end}$ 
     $| D, P, S \leftarrow \text{updateAssignments}(D, P, S, D', P', D');$ 
     $| D', P', S' \leftarrow \text{removeFinishedInteractions}(G^{ts, ts+dur}, D', P', S', ts, dur);$ 
 $\text{end}$ 
 $\text{return } D, P, S;$ 

```

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 from data. Indeed, the structure of the greedy assignment exploits the fact that the carpooling networks show a power law 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 passangerness/driverness ranking criteria.

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

- (r) random criteria ($c' = \{\text{random order}\}$, $c'' = \{\text{random order}\}$): 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 then, 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 *passengers* p and *driveness* 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 minutes starting from midnight. This produces a sliding window of length dur that moves of step 5 minutes;
- (*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).

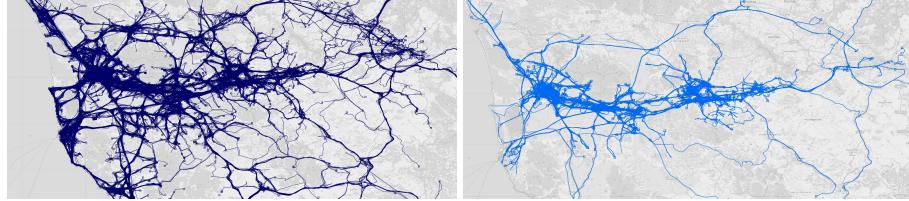


Figure 5: (Left) A sample of trajectories in Pisa province. (Right) The mobility profiles extracted.

5. Impact on Real Mobility

In this section we illustrate the results obtained from the analysis of the knowledge extracted from carpools network. First of all we report a short study related to the real mobility dataset used. Then, we describe the proportion of the amount of traffic reducible by adopting the carpools network connected with the network construction. Furthermore, we analyze the topology of the carpools networks, the ranking scores resulting from *HITS*, and we show that carpools communities are characterized by these values. Finally, we evaluate the performances of the assignment method and we report the outcomes by showing how the ranking measures and the communities help in improving the solution.

5.1. Mobility Dataset and User's Profiles

As a proxy of human mobility, we used real GPS traces collected for insurance purposes by *Octo Telematics S.p.A* [40]. This dataset contains 9.8 million car travels performed by about 159,000 vehicles active in a geographical area focused on Tuscany in a period from 1st May to 31st May 2011. Fig. 5 depicts a sample of the considered trajectories. Since the mobility dataset is geographically and temporally various, it was split following temporal and geographical principles.

In real world, different events may change how people move on the territory. Such events can be unpredictable or rare, like natural disasters, but most of them are not. The most regular and predictable event is the transition between working days and non-working days. During Saturday and Sunday, people usually leave their working mobility routines for different paths. Following this concept, we filtered out weekend trajectories maintaining only the trajectories from Monday to Friday.

Every geographical area has its own type of mobility with different characteristics depending on the surface, the topology and the number of inhabitants. In order to consider this fact, we split the dataset in provinces by considering for each province all the trajectories that pass through it. In particular, in this work we are reporting the results obtained for Pisa and Florence provinces. We selected these two cities because their mobility is quite different both in terms of number of participants and in terms of traffic flows.

Since the starting point to discover carpools interactions are the users individual routines, we performed some tests to retrieve the best parameters to extract reliable mobility profiles. The clustering algorithm used to extract the routines is a variant of

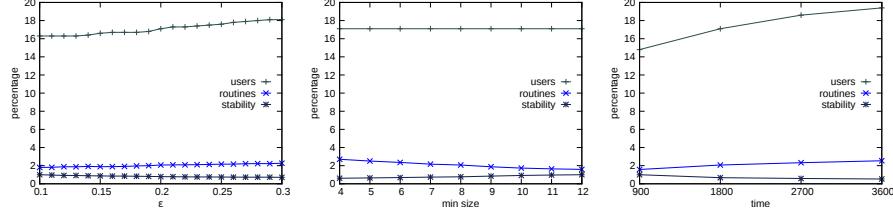


Figure 6: Profile test parameters ϵ (left), min size (center), time threshold (right)

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 in [17] for the clustering step. To tune the parameters, we have studied Optics settings on a subset of 1,000 users in Pisa dataset. In particular, we observed the following parameters. ϵ varying in range $[0.1, 0.3]$ with step 0.01, Fig. 6 (left). The bigger ϵ is, the more different trajectories are allowed to be clustered together. In other terms, it expresses the similarity allowed between trajectories. Then we studied the parameter *min size* varied in range $[4, 12]$, Fig. 6 (center). It is the minimum number of trajectories that must be in a cluster to be considered valid. Finally, we observed the time threshold *time* varying in $\{900, 1800, 2700, 3600\}$ seconds, see Fig. 6 (right). It is the max starting time difference between two trajectories and it is used by the clustering function to decide if two trajectories are synchronized. The aspects we consider to tune the values are: (i) the dataset coverage, (ii) the profile distribution per user, and (iii) the profile stability. From this empirical study we decided to use middle values because the plots obtained do not lead to a clear setting. Anyway, in each one can be found that after the middle values the curves change more rapidly than after it. For example, the time curve grows or decrease less rapidly before 1800 than after it. We choose ϵ equal to 0.2 (more or less it expresses 80% of similarity). From a temporal point of view, for our mobility dataset, a reliable value is 8 since a routine is a trajectory repeated a sufficient number of time during 20 working day. In summary, the setting used to perform the mobility profile extraction is $\epsilon = 0.2$, $minsize = 8$, $time = 30$ min.

Mobility profiles model the systematicity of each user. In Fig. 5 (right) is depicted an example of profile extracted in Pisa province. Fig. 7 (left) shows the number of routines per users in Pisa province with almost every user having one or two routines, which, should correspond to the commute to and from work. Indeed, the average number of routines per profile is 2.14, this is probably due to the home-work-home pattern. In Fig. 7 (right) is shown the temporal distribution of the trajectories and routines. Here, we can see how the profile set has a working-like trend, highlighting the three peaks during the early morning 5 - 6, lunchtime 11 - 12, and late afternoon 17 - 18. This confirms the assumption that mobility profiles are reliable to model systematic movement and thus can be exploited to retrieve shared routines among users.

5.2. Carpooling Network Analysis

In the following we analyze the characteristics of the carpooling network and we study the features used for suggesting the carpooling assignments. The power of this

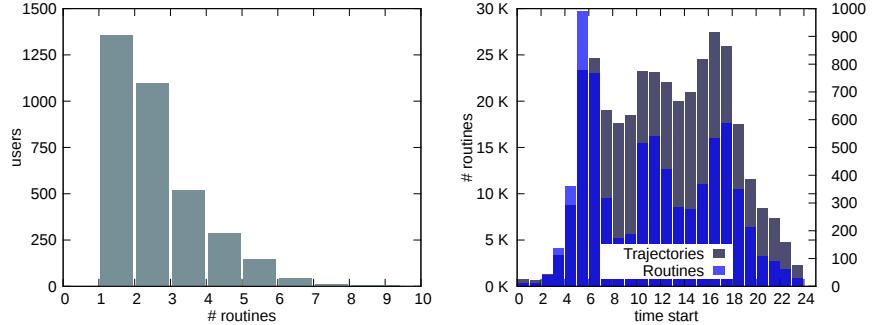


Figure 7: Routine per user distribution (left), Trajectories and routines time start distribution (right).

kind of network relies on the fact that it models and contains at the same time both mobility data and topological information. Consequently, first of all we focus on the knowledge gained from the graphs and we try to make a rough estimation of the amount of traffic reducible. Then we study the topological properties of the complex network and what can unveil ranking measures and communities.

5.2.1. Network Construction Analysis

Since the carpooling network is generated by the application of the function contained, the result depends on the value used for the parameters $spat_{tol}$ and $temp_{tol}$. Hence, in order to find good values for these parameters and to obtain a sound network made of reliable carpooling interactions, we performed a “carpooling network construction” test on a sample of 1,000 mobility profile. Fig.8 shows how the containment is affected, in percentage, in terms of routines and mobility profiles that have at least one match. It is worth to notice that by allowing a walking distance ($spat_{tol}$) of 3 km and a wasting time ($temp_{tol}$) of 30 minutes, about 60% of the profiled users have at least one match, which decreases to 10% if the walking distance becomes 500 meters. Similarly, by allowing a walking distance of 1 km and a wasting time of 60 minutes, 30% of the profiled users have at least one match, which decreases to 10% if the wasting time becomes 15 minutes. This is a clue that by enlarging the walking distance we obtain more matches than by enlarging the wasting time. Thus we built the carpooling networks for Pisa and Florence using a maximum walking distance of 1 km and a maximum wasting time of 30 minutes.

By observing the users appearing in the carpooling networks (among those which have a mobility profile), we have found that there is an high percentage of *carpoolers* (possible drivers and possible passengers). This indicates that many of them will exploit the opportunity of sharing their car the systematic traffic could significantly decrease. We can classify them as:

- *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 Divers* are the users that can act both as passenger or as drivers: $k^{out} > 0$ and $k^{in} > 0$;
- *not carpoolers* are the users that do have systematic movements but do not 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 passenger belong to PP , those which are only driver belong to PD and the users which are passenger and driver belong to both PP and PD . Fig. 9 depicts the pie chart with the normalized percentage of different type of users in the carpooling user networks of Pisa and Florence. We can observe how the carpooling potentiality appear different between the two cities. In particular, Florence seems to be more adapt than Pisa due to the larger number of passengers.

By analyzing the carpooling networks of these provinces, it is possible to have a rough evaluation of the impact that a carpooling service would have in the reduction of systematic traffic in terms of number of travels and number of cars. For example, in Pisa we extracted about 7,400 mobility routines. Supposing that every routine supports eight trajectories, that is each routine is representative of at least eight trajectories ($minsize = 8$), then the systematic mobility would be made of about 59,200 trajectories. Knowing that the routines contained in other routines, are about 1,720, then the systematic mobility would be decreased of about 23% in terms of trajectories if anyone would accept the passage for all their trajectories. We stress the fact that since the routines are repeated periodically, it is guaranteed a ride is available, or needed, every day at a certain time on that route.

We finally observe the spatio-temporal features of the routines. Fig. 10 (left) shows the routines length distributions for the categories we described above for the Pisa dataset. We notice that users who are only passenger have mainly routines with a length between 0 and 10 km, while people who are only driver have longer routines between 5 and 25 km. This fact, in conjunction with Fig. 10 (right) which shows the duration distribution, confirms in part the hypothesis that, on average, users traveling for longer distances are more likely to offer lifts, whilst users traveling shorter distances are more likely to accept lifts.

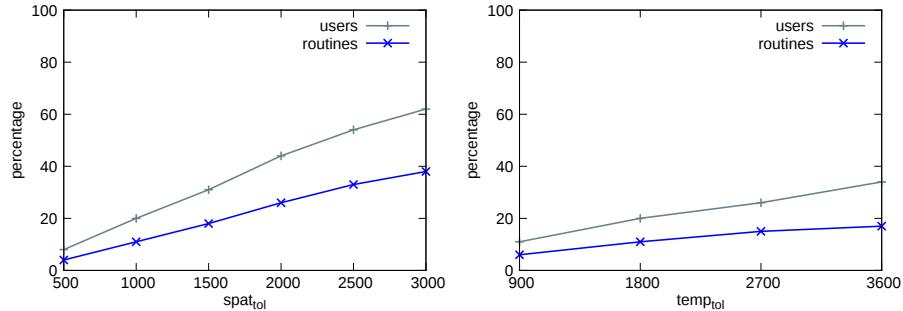


Figure 8: Carpooling network construction test of the contained parameters: (left) $spat_{tol}$, (right) $temp_{tol}$

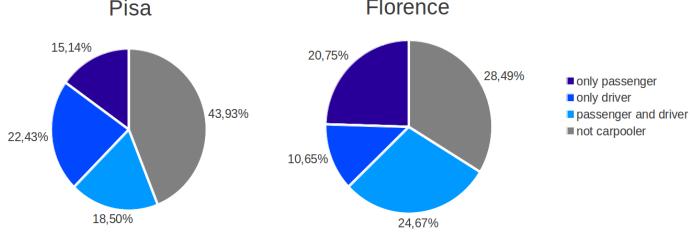


Figure 9: Carpoolers classification pie chart for Pisa and Florence.

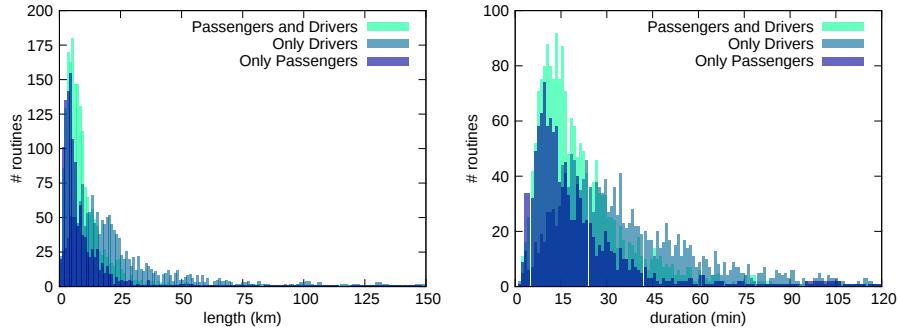


Figure 10: Routines distribution: length (left), duration (center), time start (right).

5.2.2. Topological Features Analysis

The following analysis is focused on the topological features of the carpooling user networks: the degree (in-degree k^{in} and out-degree k^{out}), and the ranking scores (driverness d and the passengerness p). The ranking scores are calculated running the HITS algorithm on the carpooling user networks (with a tolerance threshold of 1.0e-8). Fig. 11 shows both the degrees and the ranking scores distribution for Pisa and Florence. The values are normalized between zero and one in order to be comparable and the plot is in semi-log x in order to readable. Both the distributions are long tailed. This means that there are few users with high values and many users with low values. As highlighted in the previous Section there are some users, that is nodes that being only passenger or only driver have k^{in} or k^{out} equal to zero. We can notice that in Pisa province, besides the degree, many users also have a null score d and p . This is a remarkable difference between the degree and the ranking scores in the Pisa carpooling user network. In practice, despite the high correlation between k^{out} and p , and k^{in} and d , we can not state that a user that has the possibility to offer a ride to many passengers must be necessarily a *good driver* and vice-versa. On the other hand, the carpooling user network of Florence is more dense and have higher correlations with respect to degree and ranking scores.

While considering the degree the mobility behavior of the carpooling of the two cities seems similar, by observing the ranking scores we can distinguish different kinds of possible carpoolers. In Pisa the driverness d falls immediately with a steep slope

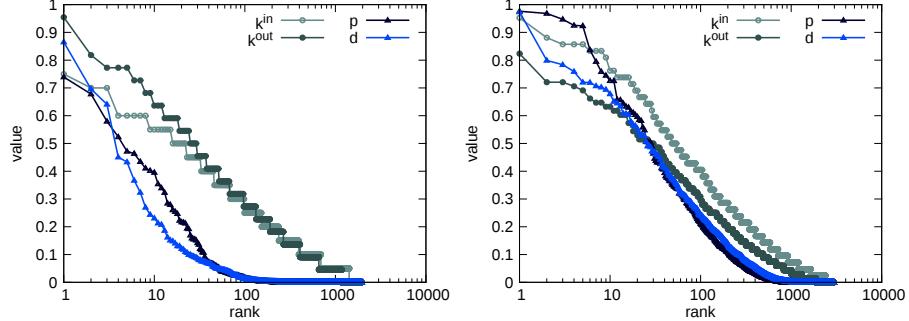


Figure 11: Degree and ranking scores distribution: (left) Pisa, (right) Florence

within the first one hundred users, while in Florence d decreases with a less steep slope within the first one hundred users, and it do not reach zero before about eight hundred users. A similar consideration can be done by looking at p . Moreover, in Pisa there are few drivers with a high d . This is an index that only few of them can serve good passengers. On the other hand in Florence there are more good drivers indicating that the shared trajectories are followed by a more relevant number of good passenger.

Furthermore, we enhance that the degree is on average very low, between 3 and 8. This is probably due to the strict parameters adopted for building the carpooling networks and to have reliable interactions. At any rate, the carpooling users networks are very sparse and this is an advantage for the task of suggesting assignment: for each user there are only few choices.

Finally, in both dataset, the standard deviations of k^{in} , k^{out} , d and p are larger than their mean. This is a sign that the users are quite heterogeneous: it does not happen that every user share a similar number of trajectories. This is probably due to the not negligible variety of systematic mobility among profiled users. The Kendall's Tau correlation coefficient applied to p and d shows that the rank between drivers and passengers is completely different (0.134): there is not any relationship between being a good driver and being a good passenger.

In Fig. 12 are shown the routines of the top ten drivers and passengers in Pisa province, that is the routines of the users with the highest d and p . These routines mainly lie between Cascina and Pontedera two cities which are close to Pisa. This result indicates that these routes are followed by many people, probably commuters.

5.2.3. Communities Analysis

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. We recall that a carpooling community is a group of users who share more routines with the users inside the community rather than with the users outside the community. Among various communities algorithm tested (Infohiermap [42], Louvain [36]) we finally selected Demon [38] because it showed the best performances in terms of time and quality of the result. Fig. 13 shows a sample of

carpooling communities in Pisa province. It is interesting to notice that the carpooling communities are well defined and geo-localized. Every community acts on a specified area that contains the systematic movements of its users. This means that a user acting in north Pisa is for sure not interested in what does a user usually moving between Cascina and Pontedera.

The topology of the sub-networks made by the communities recall the topology of the original carpooling user networks. That is, every community, from a topological point of view, behaves as the original 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). Hence the carpoolers inside a community travel along similar routes included in one another and have a similar systematic mobility.

By observing the driverness and passengerness distributions among the communities comes out that the carpooling communities can be classified in two category. Figure 14 shows the box-plot of p and d for Pisa and Florence province. We can see that some of the box-plots are clearly visible, whilst others are so little that seem invisible or are just made by outliers. This happen because some carpooling communities are made of nodes which have on average a high ranking scores, while other communities are made of nodes which have on average low ranking scores or scores close to zero. In other words, these box-plots tell us that some communities are made of nodes with considerable ranking scores, whereas other communities are made of less considerable ranking scores with respect to the overall carpooling user network. Therefore we can suppose a distinction between these two type of communities.

We decided to study what happens to the ranking scores d and p of the nodes of the communities if the HITS algorithm is performed locally to the node of every community. Basically we ran HITS inside each sub-network made by the nodes and the edges present in the carpooling community. In this way we obtained some ranking scores called *local driverness* and a *local passengerness* to distinguish them to the *global* values.

By analyzing the Kendall’s tau correlation between the global and local ranking scores for each community we have found that, in the Pisa dataset, there are about 30 communities with a correlation close to one, while the other communities (about 20 of them) have a correlation spread between zero and 0.4. Hence, there is a set of

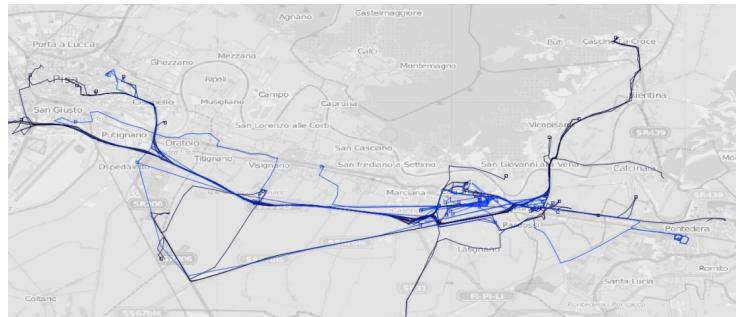


Figure 12: Top ten drivers (black) and passengers (blue) in Pisa province.

carpooling communities made by nodes which have ranking scores that are not influenced by the nodes outside the community. On the other hand, the nodes of the other communities with are influenced by inter-community links. This tells us that the first set of carpooling communities, the ones with an high Kendall's tau correlation, are *autonomous*. They are somehow stable and could rely on finding possible assignment without considering inter-community links. On the contrary, the other communities, are *not-autonomous*. They can be influenced by inter-community links and one of their user could find her potential best match with a user belonging to another community.

In Fig. 15 is reported a real example of a *not-autonomous* community (left) and an *autonomous* community (right). The ranking scores are calculated both globally (first column) and locally (second column). The bigger is a node the higher is her driverness, the more blue is a node the higher is its passengerness. Note how, according to what was exposed above, almost nothing changes for the *autonomous* community, whereas two completely different sub-networks are plotted for *non-autonomous* community. This is a visual sign of the fact that *autonomous* community are independent with respect to the whole network. They can “survive” in a carpooling scenario even if they are left alone in finding the carpooling assignments because their users are able to serve each other independently from what happens out from the community. On the contrary, a *non-autonomous* community do not have this properties and its users are influenced in a not negligible way by the others community’s users.

5.3. Carpooling Suggestions Performances

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 SOV and the number of systematic car travelling.

5.3.1. Never Drive Alone Performances

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,

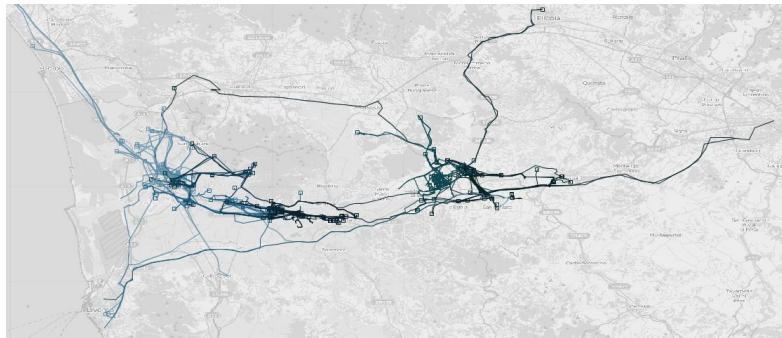


Figure 13: Geographical view of some carpooling communities in Pisa province.

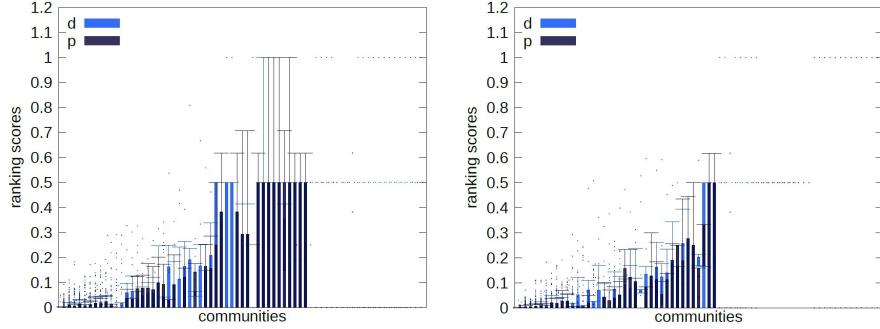


Figure 14: Carpooling ranking scores box-plot for Pisa (left) and Florence (right).

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 $dur = 1 \text{ hour}$, meaning that trips longer than one hour might be prevented from being matched to others even if the *contain* relation holds – an extremely unlikely event in our dataset, since 1-hour routines are very rare.

Figure 16 shows the percentage of passengers P^* , drivers with passengers onboard D^* and SOVs S^* for Pisa and Florence for each one combination of the criteria adopted. In addition, it shows the corresponding number of (systematic) cars on the road.

First of all, we can see that, in every case more than one third of the users become passengers, in most cases around half of them 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 results suggest that the networks considered do not lead many alternative assignment opportunities, therefore limiting the potential improvements yielded by smarter assignment criteria. This is probably due to the fact that the parameters used to build the carpooling networks, and thus to find the matching between routines, are rather strict. More tolerant settings are expected to yield even better results.

The plots show that the knowledge extracted from the mobility data and refined with network analysis progressively lead to improvement 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 turns outperforms (r).

Moreover, Figure 17 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 signal 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). At

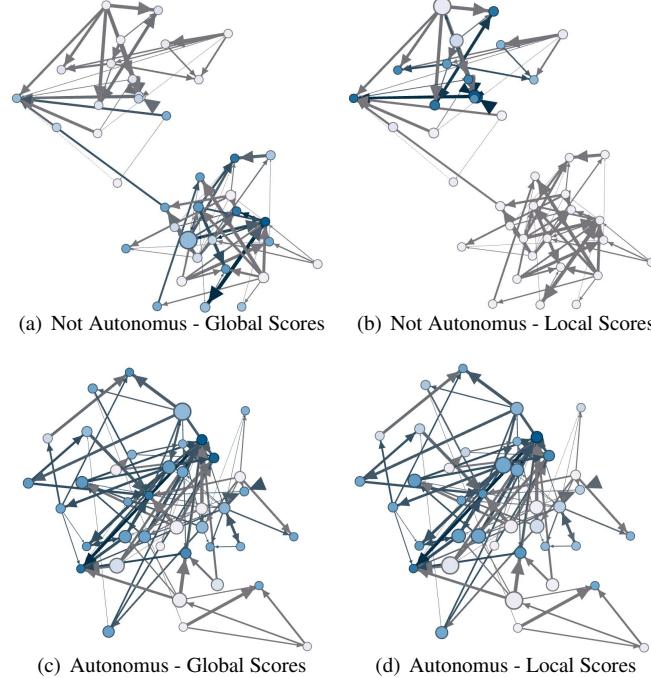


Figure 15: Network visualization of Pisa communities for not autonomous and autonomous community, and, global and local ranking scores. The bigger is a node the higher is her driverness, the darker is a node the higher is its passengerness.

any rate, the calculus with (d) is computationally more expensive, especially in periods when 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, Figure 16 also shows that, despite Florence having more good drivers and passengers than Pisa (see the carpooling user network analysis in Section 5.2), for both carpooling networks we reach comparable results in terms of suggestions.

5.3.2. Random Assignments Performances

In order to better verify that the provided solution is consistently better than those found by a random exploration of choices, we report in Figure 18 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.

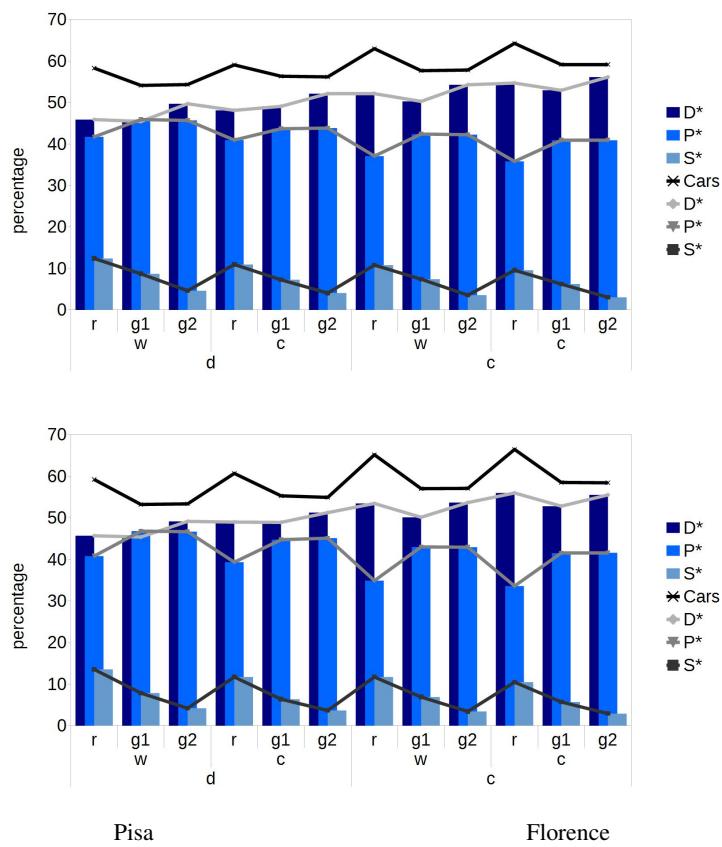


Figure 16: Assignment results for all strategies and criteria adopted: Pisa (left) and Florence (right).

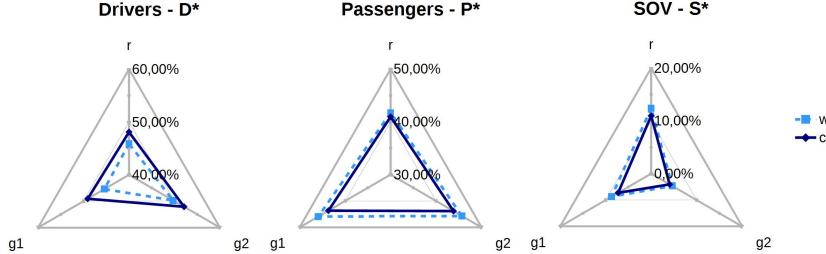


Figure 17: Assignment results for the two edge strategies (w) and (c) and for the three sorting criteria adopted on the Pisa dataset.

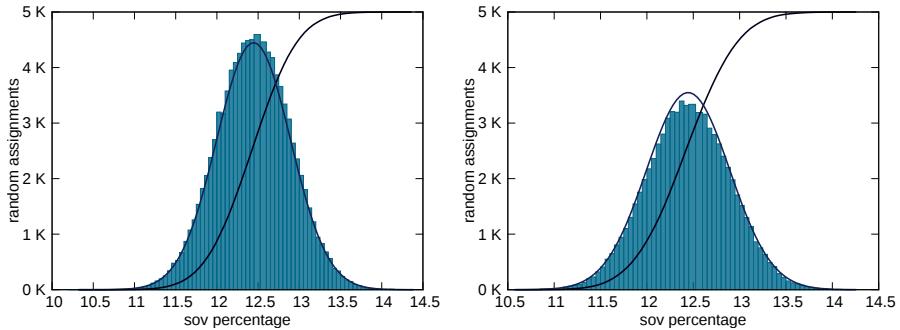


Figure 18: SOVs percentage distribution (PDF and CDF) of random assignment tests ran 100,000 times for discrete time strategies. Left: case not considering communities; right: communities are considered.

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 $6.56 \cdot 10^{-8}$, therefore very close to zero.

5.3.3. 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, also in terms of percentage over the 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).

City	# routines	# linked	# can ride	# can drive	# saved trips
Pisa	7,383	3,049	1,717	1,995	1,331 (77.52%)
Florence	9,801	5,712	3,305	4,140	2,546 (77.03%)

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).

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^3 . 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.

City	km	min	fuel(l)	€	$CO_2(kg)$
Pisa	10,868.36	24,174.58	646.67	1,001.49	1,445.49
Florence	16,748.99	43,300.28	996.56	1,543.37	2,227.62

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.

Finally, we show in Figure 19 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 and 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.

6. Conclusion and Future Work

In this paper we have proposed a novel approach for analyzing the potentiality of a carpooling service and for suggesting an assignment among systematic car drivers

³<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>

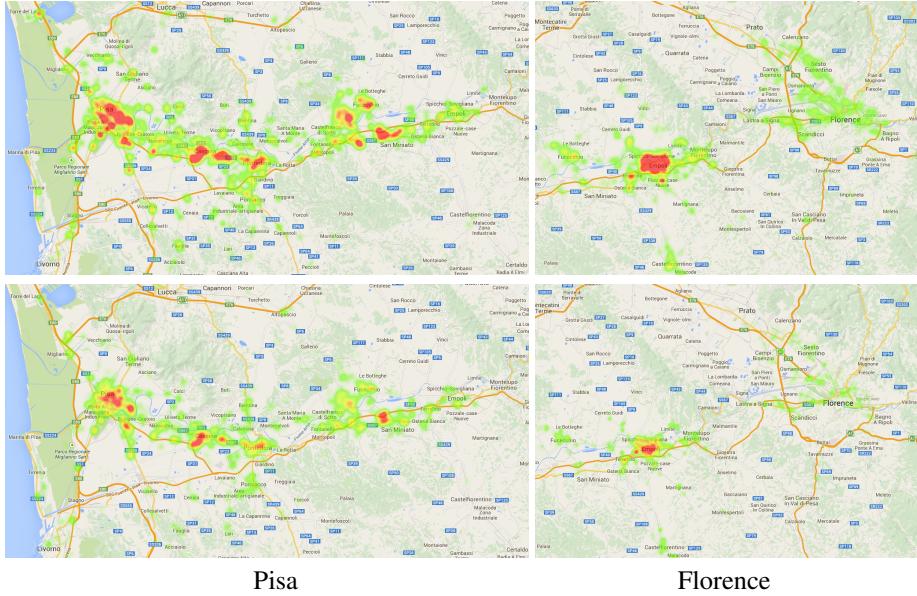


Figure 19: Spatial distribution of pick-up and drop-off points of NDA solution. Left column: Pisa province; right column: Florence province. First row: pick-up points; second row: drop-off points.

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 performed on 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 different geographical areas. Furthermore, we have found that carpooling communities can be characterized in different ways: 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 carpooling service would be proposed, then it would be better to start from autonomous communities rather than for communities less independent ones. Thus, the better candidates for a carpooling service are car drivers appearing in autonomous communities because in these communities it is easy to reduce the systematic mobility. 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 than for a short one. More-

over, 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.

Acknowledgements

This work has been partially supported by the European Commission under the FET-Open Project n. FP7-ICT-284715, ICON, and by the European Commission under the SMARTCITIES Project n. 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, *Transportation Research Part A: Policy and Practice* 32 (2) (1998) 99–114.
- [3] H. Yang, H.-J. Huang, Carpooling and congestion pricing in a multilane highway with high-occupancy-vehicle lanes, *Transportation Research Part A: Policy and Practice* 33 (2) (1999) 139–155.
- [4] J. Kwon, P. Varaiya, Effectiveness of californias high occupancy vehicle (hov) system, *Transportation Research Part C: Emerging Technologies* 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, *Regional Science and Urban Economics* 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, *Journal of Urban Economics* 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: Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, IEEE, 2010, pp. 795–800.
- [9] M. Armendáriz, J. Burguillo, A. Peleteiro, G. Arnould, D. Khadraoui, Carpooling: A multi-agent simulation in netlogo, *Proc. ECMS*.

- [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, *Procedia Computer Science* 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, *Procedia Computer Science* 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, *Transportation Research Part C: Emerging Technologies*.
- [13] S. Galland, N. Gaud, A.-U.-H. Yasar, L. Knapen, D. Janssens, O. Lamotte, Simulation model of carpooling with the janus multiagent platform, *Procedia Computer Science* 19 (2013) 860–866.
- [14] M. Armendáriz, J. Burguillo, A. Peleteiro, G. Arnould, D. Khadraoui, Carpooling: A multi-agent simulation in netlogo, *Proc. ECMS*.
- [15] 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, *Transportation Research Part C*.
- [16] R. F. Teal, Carpooling: who, how and why, *Transportation Research Part A: General* 21 (3) (1987) 203–214.
- [17] 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.
- [18] 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, *Transportation Research Part A: Policy and Practice* 45 (2) (2011) 81–90.
- [19] 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.
- [20] 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?, *Journal of Transport Geography* 22 (2012) 77–86.
- [21] 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.

- [22] K. W. Steininger, G. Bachner, Extending car-sharing to serve commuters: An implementation in austria, *Ecological Economics* 101 (2014) 64–66.
- [23] 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.
- [24] N. Bicocchi, M. Mamei, Investigating ride sharing opportunities through mobility data analysis, *Pervasive and Mobile Computing*.
- [25] 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.
- [26] 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, *Procedia Computer Science* 19 (2013) 372–379.
- [27] 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 Transactions on Algorithms (TALG)* 7 (3) (2011) 37.
- [28] R. Tijdeman, The chairman assignment problem, *Discrete Mathematics* 32 (3) (1980) 323–330.
- [29] V. Lerenc, Increasing throughput for carpool assignment matching, uS Patent App. 13/329,899 (Dec. 19 2011).
- [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, *Social Psychology Quarterly* 59 (1996) 364–374.
- [33] K. K. Charles, P. Kline, Relational costs and the production of social capital: Evidence from carpooling*, *The Economic Journal* 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] Rosvell, Infohiermap, www.tp.umu.se/~rosvall/index.html.
- [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, *Journal of algorithms* 37 (1) (2000) 146–188.
- [40] [Http://www.octotelematics.com/it](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*, Vol. 28, ACM, 1999, pp. 49–60.
- [42] M. Rosvall, C. T. Bergstrom, Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems, *CoRR* abs/1010.0431.