A Method for Managing Transportation Requests and Subdivision Costs in Shared Mobility Systems

Gianmichele Siano, Mariano Gallo and Luigi Glielmo
Universitá degli studi del Sannio, Benevento, Italy

Keywords: Shared Mobility Systems, Ridesharing, Shapley Value, Optimization Vehicle Routing.

Abstract: This paper presents an algorithm for managing the demand and supply in a shared transportation system. In particular we present a method, independent from the Geographic Information System (GIS), which processes drivers and passengers requests and ranks them in order to encourage matching and to propose the solution profitable for all. The basic idea is to give priority to the requests of passengers with more common route and avoid those with greater excess path. In the end, we propose a solution for the distribution of costs among the participants of shared travel based on the application of the Shapley value.

1 INTRODUCTION

In recent years, the business models based on shared economy and collaborative consumption are developing worldwide. Thanks to the massive use of the Internet, Smartphone and associated technologies such as GPS, GIS and Social Networks, which allow to communicate in real time and know immediately the geographical position.

Also in the transport sector systems are under development with the objective of sharing private cars among a groups of people who have a similar journey. The main aim is to promote the sustainable mobility and reduce transportation costs, traffic congestion and pollution.

In a shared transport system, typically a driver makes available to potential passengers the empty seats of his/her vehicle. To use the service, the passengers help by paying adequate costs, generally proportional to the shared journey. These systems are generally called Carpool or Ridesharing.

In this paper we will refer to real-time Ridesharing where the matching between the participants can also take a few minutes before departure or during the journey itself.

The objective of this work is to describe an optimization algorithm that can process requests from participants in the system and classify them in order to propose a solution beneficial for all.

Given a number of drivers offers and passengers requests, the problem is how to combine these demands efficiently and how to determine which of the different options are the best for each individual subject.

Since a driver and one or more passengers share different paths, another problem addressed in this paper is to define an impartial method for the subdivision of transport costs.

The study of the problems described above has been addressed in several previous studies, in particular in (Son et al., 2012) where an algorithm has been proposed "based on labeling algorithms for solving the multiobjective shortest path problem", another solution (Sghaier et al., 2010) uses an algorithm based on "Distributed Dijkstra based on the multi-agent" concept. In (Guo et al., 2012) a "Genetic Adaptive algorithm" was used, while in (Calvo et al., 2004) a system is presented using web GIS and SMS where the problem of carpooling is solved with a heuristic algorithm. Also in (Ferrari et al., 2003) a "Heuristic algorithms based on savings functions" was proposed. Finally (Santi et al., 2014) has been showed and quantified "the benefits of vehicle pooling with shareability networks".

The paper is organized as follows. In the second section we discuss the algorithm implemented in all the critical steps, then we show the temporal sequence and the communications between the partecipants, finally we propose a hypothesis of system architecture and an example of algorithm usage. In the third phase we discuss the problem of the cost distribution and propose a method based on the "Shapley value". Finally an example of application of this method is showed.

Siano G., Gallo M. and Glielmo L..
A Method for Managing Transportation Requests and Subdivision Costs in Shared Mobility Systems.
DOI: 10.5220/0005459501520158
In *Proceedings of the 1st International Conference on Vehicle Technology and Intelligent Transport Systems* (VEHITS-2015), pages 152-158
ISBN: 978-989-758-109-0

2 MANAGING TRANSPORTATION REQUESTS

2.1 Problem Description

In Figure 1 the typical representation of the problem is shown, where in a certain geographical area, there is a driver who intends to start from a position D_{start} and wants to reach the position D_{end}, given a departure time T_D. The driver has available a certain transport capacity $C_{\max}$ equal to the maximum number of seats available in the car.

In addition to the driver, in the figure there are 3 requests of passengers $(P1, P2, P3)$, with the relative position of origin and destination indicated by Pi_{start} e Pi_{end} and the respective departure times T_{Pi}.

The basic idea of the method is to define a criterion to prioritize the requests of passengers and quite reasonably we propose to avoid those with greater excess path. For excess path we mean the extra path the driver has to travel to pick up the passenger from her starting position and to accompany him to her destination.

Given two generic data points P and Q, where each point is characterized by geographical latitude and longitude, e.g. $P = (\text{lat}_P; \text{lng}_P)$, denote by $\text{length}(P,Q)$ the distance, along the path, between P e Q obtained by querying a GIS, and by $\text{dist}(P,Q)$ the distance “as the crow flies” calculated according to the formula:

$$\text{dist}(P,Q) = R \arccos[\sin(\text{lat}_P)\sin(\text{lat}_Q) + \cos(\text{lat}_P)\cos(\text{lat}_Q)\cos(\text{lng}_P - \text{lng}_Q)] \quad (1)$$

where R is the radius of the Earth. The value of dist not depends from GIS, but from the geographical coordinates transfered by drivers and passengers.

2.2 Proposed Solution

In Figure 2 we show the flowchart of the proposed algorithm, hereinafter we will be describe in detail the functionality of each block.

2.2.1 Input Block

Passing $(D_{\text{start}}, D_{\text{end}})$ to the GIS it is possible to get the driver route DR and the total travel time T_D.The effects of traffic jam are not explicitly considered, but depends by GIS utilized. DR is completely described by the set of K points that compose it through a spatial sampling:

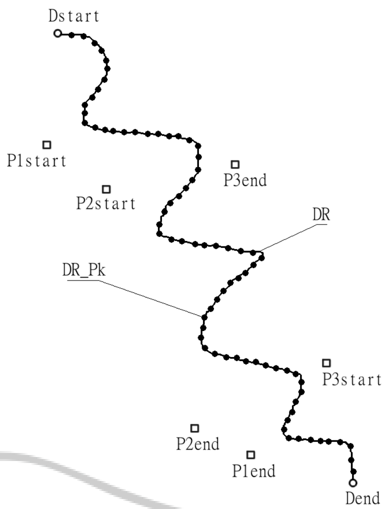

Figure 1: Typical situation.

$$DR\_P(k) \quad \text{per} \quad k = 1, \ldots, K \quad \text{where}$$

$$\begin{cases} DR\_P(1) = D_{\text{start}} \\ DR\_P(K) = D_{\text{end}} \end{cases} \quad (2)$$

Given the definitions described in the previous section inputs to our system will be:

- Driver: $D_{\text{start}}, D_{\text{end}}, DR\_P(k), T_D, C_{\max}$
- Passengers: $Pi_{\text{start}}, Pi_{\text{end}}, T_{Pi}$

2.2.2 Eligible Passengers

This block will be used to select those passengers whose requests are comparable in terms of both distance and timing.

In particular the i-th passenger, will be eligible in terms of time (time-wise) if her date and time of departure is subsequent to those of the driver. If we denote by T_D the starting time of the driver and T_{Pi} the time of departure of the passenger, the constraint will be expressed as:

$$T_{Pi} \geq T_D. \quad (3)$$

The i-th passenger will be considered eligible in terms of distance (distance-wise) if

$$\text{length}(D_{\text{start}}, D_{\text{end}}) \geq \alpha * \text{dist}(Pi_{\text{start}}, Pi_{\text{end}}) \quad (4)$$

where $\alpha \geq 1$ is a tunable parameter and

$$\text{dist}(D_{\text{start}}, Pi_{\text{start}}) + \text{dist}(D_{\text{end}}, Pi_{\text{end}}) \leq \text{length}(D_{\text{start}}, D_{\text{end}}) \quad (5)$$

Inequality (4) is used to eliminate those passengers whose route is much longer than the driver’s (e.g. $\alpha = 1.1$), while inequality (5) is important because it

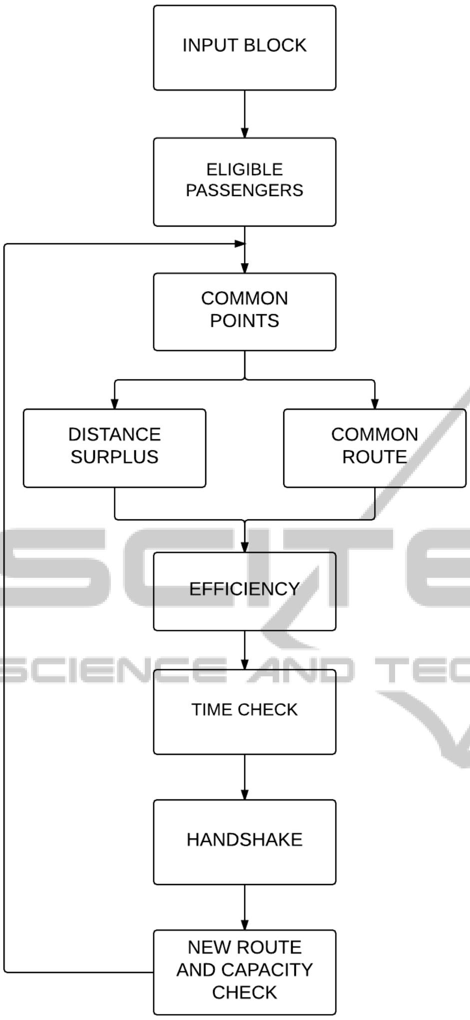

Figure 2: Flow chart.

helps to remove most passengers whose route direction is opposite to that of the driver's (e.g. passenger 3 in figure 1 would be excluded).

Once the system identified the eligible passengers, the ranking operations will start.

2.2.3 Common Points

Given all the data points associated to the geographical location of the driver $DR\_P(k)$, the first step of the loop is to determine for each eligible passenger Pi the points of the path DR with minimum distance (as the crow flies), Pi_{start} and Pi_{end}. We denote this points "Begin Common Point" $(BCPi)$ and "End Common Point" $(ECPi)$, respectively. The associated equations are

$$BCPi = DR\_P(\arg\min_j[\text{dist}(Pi_{\text{start}}, DR\_P(j))]) \quad (6a)$$

$$ECPi = DR\_P(\arg\min_j[\text{dist}(Pi_{\text{end}}, DR\_P(j))]) \quad (6b)$$

2.2.4 Distance Surplus and Common Route

Once we know the common points for the i-th passenger, you can calculate the distance in excess, $\text{dist\_}i_{\text{surplus}}$, and the length of common route $\text{length\_}i_{\text{common}}$

$$\text{dist\_}i_{\text{surplus}} = \text{dist}(P_{\text{start}}, BCPi) + \text{dist}(P_{\text{end}}, ECPi) \quad (7)$$

$$\text{length\_}i_{\text{common}} = \text{length}(BCPi, ECPi) \quad (8)$$

In Figure 3, relatively to the passenger $P1$,the distance in excess has been drawn with a dotted line with a dash-dot line.

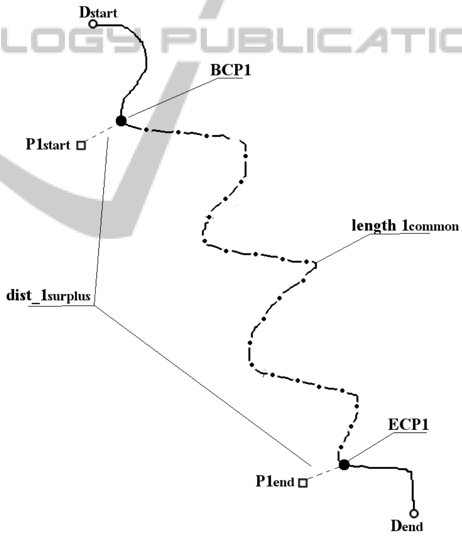

Figure 3: Distance surplus and length common.

2.2.5 Efficiency

Distances are used to sort the various passengers according to a value of efficiency:

$$\text{Eff}_i = \frac{\text{length\_}i_{\text{common}}}{\text{length\_}i_{\text{common}} + \gamma \text{dist\_}i_{\text{surplus}}} \quad (9)$$

This value Eff_i favors passengers who have long common route and short surplus distance. The value of γ depends on the preferences of the driver, who decides what weight to attribute to route excess compared to the common one.

2.2.6 Time Check

Afterwards the algorithm checks whether the driver arrives at the starting point of the passenger later than the desired starting time of the passenger; this condition can be approximately checked as

$$T_{Pi} \leq T_D + T_{\mathrm{DR}} \left(\frac{\arg[BCPi]}{K} + \frac{\mathrm{dist}(BCPi, Pi_{\mathrm{start}})}{\mathrm{length}(D_{\mathrm{start}}, D_{\mathrm{end}})} \right) \quad (10)$$

2.2.7 Handshake

In this block, the system notifies the driver with a sorted list according to the metrics described above; then, the system awaits the decisions of the driver and passenger. The step of decision and agreement will be described later (subsection 2.4).

2.2.8 New Route and Capacity Check

If the actors reach an agreement, the passenger is added to the shared travel and the capacity is decremented by one.

A new request is made to the GIS for the exact route calculation, all associated values and resulting rescheduling times. This new request takes into account the passage through the begin and end points of the chosen passenger.

In Figure 4 we show how the route is changed after addition of the passenger $P1$.

This operation will be repeated recursively until either the driver does not detect any potential passenger or the numbers of places available si zero.

2.3 Example

In this example we show the operations of the algorithm described above where $\gamma = 4$, in this case for simplicity the time variable is not considered. For the driver:

- $D_{\mathrm{start}} = \mathrm{Napoli}(40.8517763, 14.2681383)$
- $D_{\mathrm{end}} = \mathrm{Milano}(45.4654323, 9.1859402)$
- $\mathrm{length}(D_{\mathrm{start}}, D_{\mathrm{end}}) = 773,74$ km
- $K = 12651$

In (DRPK, 2014) you can find (lat,lng) data of $DR\_P(k)$ of the route (Napoli,Milano). These data are found using GIS (GoogleMaps, 2014).

Further, we constructed a list of hundred of passengers randomly generated. For every passenger the towns of departure and arrival with the relative geographic locations are known:

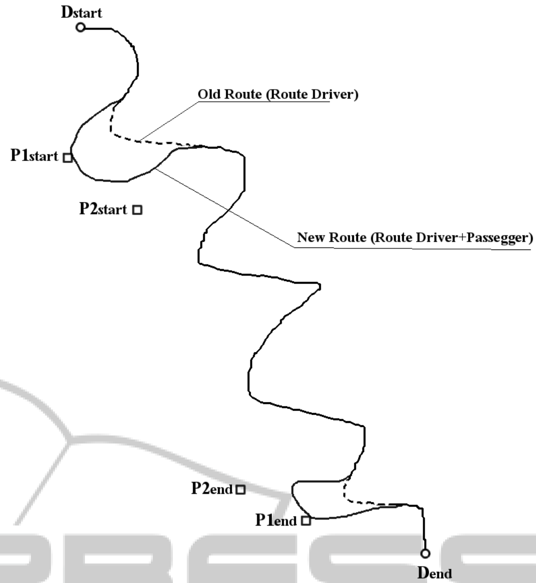

Figure 4: Old Route and new Route.

- $i, (\mathrm{lat}_{P_{\mathrm{start}}}, \mathrm{lng}_{P_{\mathrm{start}}}), (\mathrm{lat}_{P_{\mathrm{end}}}, \mathrm{lng}_{P_{\mathrm{end}}})$

This information is in (data.cvs, 2014)

The Table 1 below shows the first 4 passengers of the sorted list.

Table 1: First four passengers.

N	i	BCP	ECP	Eff
P_1	10	44.473,11.270	45.349,9.310	0.960
P_2	32	40.854,14.318	43.779,11.161	0.801
P_3	81	40.869,14.321	45.168,9.598	0.792
P_4	88	41.996,12.671	44.524,11.151	0.756

2.4 Time Sequence

Figure 5 showns the time sequence of the algorithm described above.

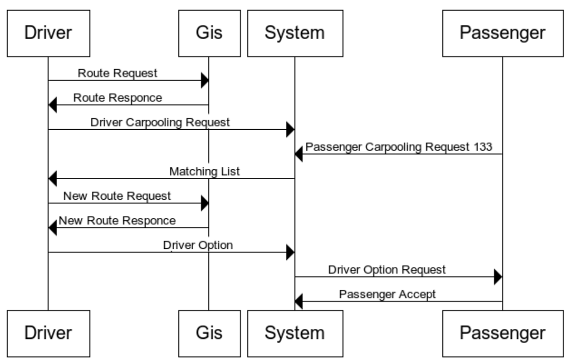

Figure 5: Time Sequence.

The driver initially, known geographic locations of departure D_{start} and destination D_{end}, sends a request to the GIS (Route Request), which responds by transmitting information about the different possible routes (Route Response). Each route contains the travel path DR (Driver Route), the length “length$(D_{\text{start}}, D_{\text{end}})$”, the travel time T_D and all points $DR\_P(k)$.

The driver selects the most appropriate route and informs the central system which performs the algorithm. The system processes the driver request (Driver Carpooling Request).

At a later time a generic passenger i with geographic location of departure Pi_{start} and desired destination Pi_{end} sends its own request to the central system (Passenger Carpooling Request).

The system evaluates the different requests of passengers and transmits to the driver the sorted list (Matching List) based on the algorithm previously described.

When a passenger has been chosen the driver sends a new request to the GIS (New Route Request). The GIS replies with a new route, that starts from the starting point of the driver, passes through the points of origin and destination of the passenger and ends with the destination of the driver. (New Route Response).

Comparing information between old and new path it is possible to know with precision the true path excess, the true common path and the economic benefits.

This option, with the corresponding economic values, is communicated to the System (Driver Option). The System notifies the passengers (Driver Option Request) who decides whether accepting the proposal or not (Passenger Accept).

2.5 Architecture

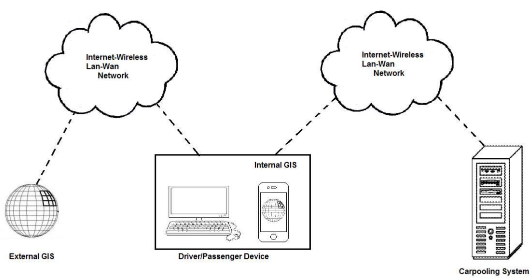

Figure 6: Architecture.

Figure 6 shows a possible system architecture. The devices (Driver/Passenger Device) that interact with the system can be Smartphone, PC, PDA, Tablet or other. Some of these systems may have an Internal GIS others instead require an internet connection to send requests to gis (External GIS).

Note that in this architecture the system that manages the different request of the drivers and passengers does not require direct communication with the GIS.

3 COST SUBDIVISION

3.1 Problem Description

In shared trasportation system an important problem is how to evaluate a fair division of the costs as function of the journeys in common among the various participants.

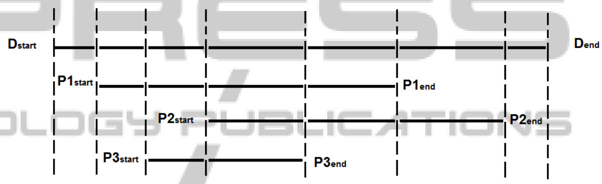

Figure 7: Different shared paths.

Figure 7 is shows an example where a driver and three passengers share different paths

For the solution of the problem we used a method based on the “Shapley value” (Osborne and Rubinstein, 1994), which given a coalition and an associated payoff redistributes the payoffs in proportion to the contribution that each player gives to the coalition. An important property of the Shapley value is that it considers the order of the player joining the coalition in computing their respective contribution to the “coalition”.

3.2 Shapley Value

In a shared transportation system the model of the driver and passengers is comparable to a cooperative game, with transferable utility and superadditivity.

Given N, the set of players (driver D and passengers Pi) with $|N| \geq 2$ and S, R two generic coalitions $S, R \subseteq N$, define v the characteristic function, i.e. the cost coalition S has to pay to make the shared journey.

Given a path length d_{route} and a travel time t_{route} we can calculate the cost associated approximately as:

$$c_{\text{route}} = \beta_1 * d_{\text{route}} + \beta_2 * t_{\text{route}} \tag{11}$$

where β_1 is a coefficient that considers the type of route and β_2 considers the time spent for moving (in Italy, β_1 is variable from a minimum of 0.06 €/km for motorways to a maximum of 0.12 €/km for city driving and β_2 is 15-20 €/h for working time and 7 €/h for holiday time).

Thus in our case the characteristic function has the following properties, required to apply Shapley's method:

$$\begin{cases} v(\emptyset) = 0 \\ v(S \cup R) < v(S) + v(R) \text{ if } D \in S \text{ or } D \in R \\ v(S \cup R) = v(S) + v(R) \text{ otherwise} \end{cases} \quad (12)$$

The cost of coalition composed by only passengers is a sum of the cost associated to the single passenger. In other words coalitions including a driver and passengers have a total cost which is less than the sum of subcosts; if, instead, neither coalition includes a driver, the total cost is the sum of costs.

The Shapley value is used for distributing the total cost of the coalition among its members and aims to subdivision in proportion to the marginal cost that each player adds to the coalition.

First we need to calculate all the possible orderings of N elements and then we make an average of all the marginal costs of the individual player on all orderings previously calculated. The value of N is incremented by one every time we add a new passenger to the coalition.

This value is calculated using:

$$\phi(i) = \frac{1}{|N|!} \sum_{\pi \in \Pi_N} [v(B(\pi, i) \cup \{i\}) - v(B(\pi, i)] \quad (13)$$

where:

- $\phi(i)$ Cost of the i-th participant to shared travel;
- Π_N Set of all possible orderings of the elements of N (permutations);
- $B(\pi, i)$ is the set of players in N which precede the player i in the ordering considered.

3.3 Example

Suppose there is a driver and two passengers, the total cost is calculated using (11) where β_1 = 0.09 €/km and β_2 = 0 €/h. For the computation of d_{route} using (GoogleMaps, 2014).

Table 2 summarizes the input data.

Table 2: Example 1: Input Data.

Users	Begin	End	Cost €
D	Napoli	Milano	69.57
P_1	Caserta	Bologna	49.68
P_2	Roma	Firenze	25.02

Table 3: Example 1: Characteristic function.

User	Cost €
D	69.57
$P1$	49.68
$P2$	25.02
$D, P1$	72.27
$D, P2$	72.63
$P1, P2$	74.70
$D, P1, P2$	75.33

Table 4: Example 1: Solution.

User	Shared cost €	Saving
D	35.1	49.55 %
P_1	26.19	47.28 %
P_2	14.04	43.88 %

Using the input data we can build the characteristic function, respecting (12). The Table 3 shows the associated cost for each coalition.

Table 4 reports the costs distributed according to Shapley's value using (13).

Another example of cost calculation using this method and the data in (solution.cvs, 2014) is shown in Table 5.

Table 5: Example 2: Solution.

User	Initial cost	Shared cost €	Saving
D	69.57	9.19	86.8 %
P_1	45.0	24.85	44.77 %
P_2	17.28	9.13	47.19 %
P_3	38.16	21.13	41.9 %
P_4	36.36	21.13	41.9 %

4 CONCLUSIONS

In this article we studied the problem of shared transport system, in particular we proposed an algorithm that can process requests from players in the system and rank them in order to propose the solutions beneficial for all.

The algorithm implemented favors the demands of passengers with longer route in common and avoid those with greater excess path. The algorithm is independent of GIS, because the system works with the GIS data requested by driver.

Finally a solution to the problem of cost-sharing among participants in a shared transport system based on the application of Shapley's value was exposed.

For this method, a patent application was presented (Siano et al., 2014).

REFERENCES

Calvo, R., de Luigi, F., Haastrup, P., and Maniezzo, V. (2004). A distributed geographic information system for the daily carpooling problem. *Computers and Operations Research V.31, I.13, pp. 2263-2278*.

data.cvs (2014). https://drive.google.com/file/d/0B_mFYm1leYv9Q05SN0g4bzdmdjA/view?pli=1.

DRPK (2014). https://drive.google.com/file/d/0B_mFYm1leYv9OHRzYU9qOFJLUnM/view?usp=sharing.

Ferrari, E., Manzini, R., Pareschi, A., Persona, A., and Regattieri, A. (2003). The car pooling problem: Heuristic algorithms based on savings functions. *Journal of Advanced Transportation V.37, I.3, pp. 243-272*.

GoogleMaps (2014). https://www.google.it/maps/dir/Napoli/Milano.

Guo, Y., Goncalves, G., and Hsu, T. (2012). A multi-agent based self-adaptive genetic algorithm for the long-term carpooling problem. *Journal of Mathematical Modelling and Algorithms in Operations Research, V.12, I.1, pp. 45-66*.

Osborne, M. and Rubinstein, A. (1994). *A course in Game Theory*. The MIT Press, Cambridge, Massachusetts.

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S., and Ratti, C. (2014). Quantifying the benefits of vehicle pooling with shareability networks. *PNAS V.111, pp. 13290-13294*.

Sghaier, F., Zgaya, S., Hammadi, T., and Tahon, F. (2010). A distributed dijkstra's algorithm for the implementation of a real time carpooling service with an optimized aspect on siblings. *IEEE Annual Conference on Intelligent Transportation Systems*.

Siano, G., Gallo, M., and Glielmo, L. (2014). Algoritmo per la gestione di un sistema di carpooling. *N. BN2014A000012, CCIAA di Benevento Cod. 62*.

solution.cvs (2014). https://drive.google.com/file/d/0B_mFYm1leYv9NkxnWkNUeGdfa2M/view?usp=sharing.

Son, T., An, L., Tao, P., and Khadraoui, D. (2012). A distributed algorithm solving multiobjective dynamic carpooling problem. *International Conference on Computer Information Science*.