

On the Many-to-Many Carpooling Problem in the Context of Multi-Modal Trip Planning*

Lorenzo Ricciardi Celsi, Alessandro Di Giorgio, Raffaele Gambuti, Andrea Tortorelli, and Francesco Delli Priscoli

Abstract— In this paper an approach to the many-to-many carpooling problem with automated passenger aggregation is presented. The proposed solution allows to optimally solve the related routing problem, by relying on a constrained shortest path algorithm, for users travelling within multiple transportation networks, thus enabling multi-modality, and exploits the users' availability to be aggregated into carpools. The mathematical model behind the proposed approach is illustrated. Then, an algorithmic procedure capable of reasonably coping with the complexity that arises in real-sized scenarios, often characterized by multiple heterogeneous data sources, is discussed. Finally, simulations are reported in order to evaluate the effectiveness and the performance of the proposed approach.

I. INTRODUCTION

Over the years, many countries have been implementing policies aimed at reducing the harmful emissions caused by ground transportation [1]-[3]. In particular, since urban contexts are very often characterized by the presence of some attractive points (e.g., schools, offices, shopping centers), it is likely to have groups of people moving along the same path. This, together with the fact that most cars carry the driver only [4], has suggested the potential positive impact of a rationalization of urban traffic flows. In this respect, several solutions can be found in the literature, all of which involve the aggregation of users' trips either at public level (i.e., *Demand Reactive Transportation*, DRT [5]) or at private level (i.e., *carpooling*). This paper is focused on carpooling as a key enabler of multi-modal trip planning in urban scenarios. As specified in [5], tackling the problem of trip aggregation at private level allows to remove structural costs while promising lower prices for end users, if compared with DRT.

The most general formulation of the carpooling problem involves (i) a group of users whose role (driver or passenger) should be determined, (ii) the composition of the pools and (iii) the related routes, possibly in real time. As discussed in [6], such a problem is NP-hard and thus it is generally tackled by means of (meta-)heuristics. More precisely, carpooling algorithms can be classified based on the presence or absence of one of the following characteristics:

1. users share either the same source or the same destination (i.e., the so-called *go-to* and *return-from*

work problems, respectively);

2. the role of each user is known in advance;
3. users cover the same trip periodically;
4. apart from going by car, only one mode of transportation is allowed;
5. users can specify their priorities/preferences.

It goes without saying that carpooling algorithms are also classified according to the nature of the matching procedure (manual/automated) and to the reference time scale (whether they be designed to work in real time or on a day-ahead basis). While points 1-3 simplify the problem, point 5 certainly increases the problem complexity. Moreover, point 4 connotes a more general service which, along with the aggregation of users, is also capable of planning a *multi-modal* trip.

In the literature, the carpooling problem is usually formalized either as a *many-source single-destination* (MS-SD) problem (i.e., the go-to work one) or as a *single-source many-destination* (SS-MD) problem (i.e., the return-from work one) [7]-[12]: both cases involve less computational costs than the general *many-to-many* (M2M) case. Nonetheless, while the literature related to the MS-SD and SS-MD carpooling problems is sufficiently developed, the literature related to the M2M carpooling problem is still poor in efficient solutions. Also, most of the proposed solutions are non-automated. In other words, the actual creation of carpools is left to users: they are provided with nothing but a platform allowing them to communicate their trip and/or search for suitable pools. Another aspect that should be taken into account is the optimality of the carpool path. Indeed, most approaches consider the passenger sources and destinations as fixed. However, such an assumption does not allow to yield optimality whereas the automated bending of both driver's and passengers' trajectories allows to increase the number of carpools that can be arranged and to reduce the impact of carpooling, in terms of time and cost, on the user's trajectory.

The paper is organized as follows. Section II presents the mathematical formulation of the M2M carpooling problem with automated passenger aggregation within an urban area, which is solved by the multi-modal trip planning functions embedded in the BONVOYAGE Intelligent Transportation System (ITS, see the H2020 project that goes under the same name [13]). Section III discusses the heuristic algorithm adopted for solving the considered problem. Eventually, the

* Research supported by the European Commission in the framework of the H2020 BONVOYAGE project (*From Bilbao to Oslo, intermodal mobility solutions and interfaces for people and goods, supported by an innovative communication network*) under Grant Agreement n° 635867.

L. Ricciardi Celsi, A. Di Giorgio, R. Gambuti, A. Tortorelli, and F. Delli Priscoli are with the Department of Computer, Control, and Management Engineering Antonio Ruberti, Università degli Studi di Roma La Sapienza, via Ariosto 25, 00185 Roma, Italy (email: ricciardicelsi@diag.uniroma1.it).

effectiveness of the approach is shown through simulation results and possible improvements are discussed.

II. MATHEMATICAL FORMULATION

The goal of this section is to derive a mathematical model for the carpooling problem with automated passenger aggregation and characterize the set of possible solutions to such a problem in the M2M scenario. Due to the problem complexity, the approach followed is structured into the following two steps.

Step (1) At first, passenger trajectories are computed over a multi-layer network including public and pedestrian networks as well as the portion of the private network identified by each driver's path along with his/her detour range. If there happen to be passengers for whom it is convenient to use the private network and hence join a carpool, their *entry* and *exit points* are stored and will be used in the next step of the algorithmic procedure. Entry and exit points are defined as those nodes of the private network where passengers should be picked up and dropped off, respectively, by the driver.

Step (2) Then, the routing problem from the drivers' point of view is tackled: as a consequence of the previous step, the problem reduces to the computation, for each driver, of the shortest path through fixed points (i.e., the entry and exit points for those passengers who are forming the carpool).

Before tackling the routing problem from the passengers' and drivers' viewpoints (Section II.B), in the following subsection a mathematical description of the transportation networks used is provided.

A. Network Modelling

In an urban context, roads can support different types of users such as pedestrians, bicycles and private/public vehicles. Hence, each transportation mode is associated with the corresponding transportation network, which can be modelled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with \mathcal{V} and \mathcal{E} being respectively the sets of graph vertices and edges. In this work, only three modes of transportation are considered – namely, private cars, public vehicles and walking – but the extension to other modes is straightforward. As a result, the following graphs are defined:

- $\mathcal{G}_{\text{pri}} = (\mathcal{V}_{\text{pri}}, \mathcal{E}_{\text{pri}})$, modelling the private transportation network;
- $\mathcal{G}_{\text{pub}} = (\mathcal{V}_{\text{pub}}, \mathcal{E}_{\text{pub}})$, modelling the public transportation network;
- $\mathcal{G}_{\text{ped}} = (\mathcal{V}_{\text{ped}}, \mathcal{E}_{\text{ped}})$, modelling the pedestrian network.

The routing problem can be solved on one or more of these graphs, depending on the user type. Moreover, we assume that passengers can specify if they want to be aggregated with other users (namely, with drivers) and/or if they want to use public vehicles. In both cases, passengers' trajectories belong to different transportation networks. On the other hand, drivers can travel only over \mathcal{G}_{pri} and thus their paths can be computed with classical shortest path algorithms: once the path for the d -th driver has been computed, the sub-graph $\mathcal{G}_{\text{pri}}^{(d)} \subseteq \mathcal{G}_{\text{pri}}$, modelling the portion of \mathcal{G}_{pri} that the d -th driver

is available to sweep for picking up passengers, can be defined based on the detour range specified by the driver him/herself. This means that the aggregation procedure, from the driver's viewpoint, considers $\mathcal{G}_{\text{pri}}^{(d)}$ and not \mathcal{G}_{pri} . Instead, as regards passengers, since multiple transportation modes are allowed, the routing problem relies on a more complex network structure: namely, for each passenger the following *multi-graph* is defined:

$$\mathcal{G}_p^{(d)} = \mathcal{G}_{\text{ped}} \cup \mathcal{G}_{\text{pub}} \cup \mathcal{G}_{\text{pri}}^{(d)}$$

i.e., $\mathcal{G}_p^{(d)}$ models the network over which passengers can travel, including not only the pedestrian and public networks but also the portion of the private network identified by the d -th driver (more precisely, by his/her source node, destination node and detour range).

B. Carpooling Problem Formulation

As anticipated, in order to solve the M2M carpooling problem with automated passenger aggregation, a procedure structured into two steps is presented. In particular, in Step (1), it is checked if there are passengers eligible for aggregation and then if there are drivers with whom they can be matched. The problem, from the point of view of each passenger, consists in determining the shortest path over the multigraphs $\mathcal{G}_p^{(d)}$ for each $d = 1, \dots, M$, where M is the number of available drivers. If the solution path does not include a sub-path on the private network $\mathcal{G}_{\text{pri}}^{(d)}$, this means that for the considered passenger it is not convenient to share part of the trip with the d -th driver. Conversely, if the passenger's solution path implies that part of the trip occurs on $\mathcal{G}_{\text{pri}}^{(d)}$, then it means that the considered passenger is eligible to be aggregated with the d -th driver. If this is the case, the nodes where he/she is meant to be picked up and dropped off (i.e., the first and last nodes of the passenger's sub-path over $\mathcal{G}_{\text{pri}}^{(d)}$) are stored as the corresponding entries of two arrays S and D , respectively. Hence, after this procedure has been carried out for each passenger, these two sets S and D contain all the sources and destinations of passengers, with respect to the private network, that is, they identify the points through which the d -th driver should pass in order to create a carpool. More in detail, Step (1) can be formulated as a single-objective multi-modal *Shortest Path Problem* (SPP) where the function to be minimized is the total cost along the path. This instance of the SPP has gained popularity in recent years due to the development of ITSs. The solution method adopted in this work is based on [14] and consists in a modified version of Dijkstra's algorithm, referred to as the Generalized Dijkstra's (GD) algorithm.

Then, Step (2) tackles the routing problem from the driver's point of view: such a problem consists in determining the minimum-cost path, passing through a set of fixed points, for each driver (who identifies each carpool). In other words, the aim is to determine the best sequence of actions in the set $A := \{\text{pick up passenger } 1, \dots, \text{pick up passenger } N, \text{drop off passenger } 1, \dots, \text{drop off passenger } N\}$. In this respect, a linear programming formulation, designed as an extension to the M2M case of the solution procedure presented in [12] is proposed.

TABLE I. VARIABLES AND PARAMETERS USED IN STEP (2)

N	Number of passengers
M	Number of drivers
S, D	Ordered sets of entry and exit points, respectively, of all passengers
Σ, Δ	Ordered sets of sources and destinations, respectively, of all drivers
$\eta = \{S \cup D \cup \Sigma \cup \Delta\} \subseteq \mathcal{V}$	Ordered set including all entry points, exit points, sources and destinations
x_{ij}	Polygonal chain (or polyline) connecting node $v_i \in \eta$ to node $v_j \in \eta$
$w_{ij}, l_{ij}, \tau_{ij}$	Cost, length and travel time of the path identified by x_{ij}
χ_{ijkd}	Boolean decision variable
A_d	Number of free seats made available by the d -th driver
K_d	Number of consecutive polylines composing the d -th driver's solution path
L_d	Maximum allowed length for the d -th driver's solution path
T_d	Maximum allowed time for the d -th driver's solution path
α_j	Maximum allowed waiting time for passenger j
γ_j	Time when passenger j arrives at its pick-up point
β_d	Maximum allowed waiting time for the d -th driver
ζ_j	Maximum allowed time when passenger j should be dropped off

Table I describes the variables and parameters appearing in the mathematical formulation of Step (2). It is also assumed that the set η containing sources and destinations of all users is ordered in the sense that the elements $v_j, v_{N+j} \in \eta$ are, respectively, the entry and exit points of the same user $j = 1, \dots, N$ (i.e., a passenger), whereas $v_{2N+d} \in \eta$ and $v_{2N+M+d} \in \eta$ are the source and destination of driver $d = 1, \dots, M$. In addition, assuming that variables $\alpha_j, \gamma_j, \zeta_j$, and $v_j, v_{N+j} \in \eta$ all refer to the same passenger j and, similarly, that L_d, T_d, K_d and β_d refer to the same driver d , the solution path for the d -th driver can be identified by a sequence of K polylines (i.e., x_{ij}). Let χ_{ijkd} be a Boolean decision variable defined as

$$\chi_{ijkd} = \begin{cases} 1 & \text{if the } k\text{-th polyline of the} \\ & \text{solution path of the } d^{\text{th}} \text{ driver} \\ & \text{coincides with } x_{ij}, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $k = 1, 2, \dots, (K-1)$ represents each moment when a passenger is either picked up or dropped off, while $k = K$ represents the instant when the d -th driver reaches his/her destination. Furthermore, consider the set of Boolean auxiliary variables y_{pd} defined as

$$y_{pd} = \sum_{i=1}^{2N+1} \sum_{k=1}^{K-1} \chi_{ipkd} \quad (2)$$

i.e., $y_{pd} = 1$ if the p -th passenger is aggregated with the d -th driver. With the above notation in mind, the drivers' routing problem can thus be formulated as follows:

$$\min \sum_{d=1}^M \sum_{i=1}^{2N+2M} \sum_{j=1}^{2N+2M} \sum_{k=1}^K w_{ij} \chi_{ijkd} \quad (3)$$

subject to

$$\sum_{j=1}^N \chi_{(2N+d)j1d} + \chi_{(2N+d)(2N+M+d)1d} = 1 \quad (4)$$

$$d = 1, \dots, M,$$

$$\sum_{h=N+1}^{2N} \chi_{h(2N+M+d)kd} + \chi_{(2N+d)(2N+M+d)kd} = 1 \quad (5)$$

$$d = 1, \dots, M,$$

$$\sum_{i=1}^{2N+M} \sum_{k=1}^K \chi_{ijkd} - \sum_{i=1}^{2N+M} \sum_{k=1}^K \chi_{i(j+N)kd} = 0 \quad (6)$$

$$j = 1, \dots, N, \quad d = 1, \dots, M,$$

$$\sum_{i=1}^{2N} \chi_{ij\tau d} - \sum_{i=1}^{2N+M} \sum_{k=1}^{\tau-1} \chi_{(j-N)ikd} = 0 \quad (7)$$

$$j = 1, \dots, N, \quad \tau = 1 \dots, K-1, \quad d = 1, \dots, M,$$

$$\sum_{j=1}^{2N+2M-1} \sum_{k=1}^K \chi_{ijkd} \leq 1 \quad (8)$$

$$i = 1, \dots, 2N+2M-1, \quad d = 1, \dots, M,$$

$$\sum_{i=1}^{2N+M} \sum_{k=1}^r \chi_{ijkd} - \sum_{i=1}^{2N+M} \chi_{ji(r+1)d} = 0 \quad (9)$$

$$r = 1, \dots, K-1, \quad d = 1, \dots, M, \quad j = 1, \dots, 2N+M$$

$$\sum_{i=1}^{2N+M} \sum_{j=1}^{2N+2M} \chi_{ijkd} = 1, \quad k = 1, \dots, K \quad (10)$$

$$d = 1, \dots, M$$

$$\left| \sum_{i=1}^{2N+M} \sum_{j=1}^{2N} \tau_{ij} \chi_{ijkd} - \gamma_p \right| \leq \min\{\alpha_p, \beta_d\} \quad (11)$$

$$p = 1, \dots, N, \quad d = 1, \dots, M, \quad k = 1, \dots, K-1$$

$$\sum_{i=1}^{2N+M} \sum_{j=1}^{2N} \sum_{k=1}^r \tau_{ij} \chi_{ijkd} \leq \zeta_p \quad (12)$$

$$r = 1, \dots, K-1, \quad d = 1, \dots, M, \quad p = 1, \dots, N$$

$$\sum_{i=1}^{2N+M} \sum_{j=1}^{2N+2M} \sum_{k=1}^K \tau_{ij} \chi_{ijkd} \leq T_d \quad (13)$$

$$d = 1, \dots, M,$$

$$\sum_{i=1}^{2N+M} \sum_{j=1}^{2N+2M} \sum_{k=1}^K l_{ij} \chi_{ijkd} \leq L_d \quad (14)$$

$$d = 1, \dots, M,$$

$$\sum_{j=1}^M y_{ij} \leq 1, \quad i = 1, \dots, N \quad (15)$$

$$\sum_{i=1}^N y_{ij} \leq A_j, \quad j = 1, \dots, M \quad (16)$$

where

- (3) specifies that the total drivers' routing cost is the objective function to be minimized;
- (4) specifies that the computed routes have to start at the drivers' sources;
- (5) specifies that the computed routes have to end at the drivers' destinations;
- (6) specifies that a passenger who has been picked up at some node should be left at some other node;

- (7) specifies that a passenger cannot be dropped off before being picked up;
- (8) specifies that, once in a node, only one polyline can be chosen at the next iteration;
- (9) specifies that the destination and source node of, respectively, the k -th and $(k + 1)$ -th polylines coincide;
- (10) specifies that at each k exactly one polyline is chosen;
- (11) specifies that driver d (for $d = 1, \dots, M$) will not wait at the entry point of passenger p (for $p = 1, \dots, N$) more than $\min\{\alpha_p, \beta_d\}$;
- (12) specifies that a passenger is dropped off at his/her destination within his/her required time;
- (13) specifies that the driver has to reach his/her destination within his/her required time;
- (14) specifies that the driver's path should not be longer than the maximum length he/she has previously indicated;
- (15) specifies that each passenger can be aggregated with only one driver;
- (16) specifies that, at most, as many passengers as the maximum available seats are aggregated with each driver.

In particular, also in Step (2), as optimality criterion for the driver we choose to minimize the overall travel cost only. The authors are currently working on the implementation of more complex personalized optimality criteria, weighing different objectives, both for passengers (in Step (1)) and drivers (in Step (2)).

Moreover, for the sake of simplicity, in this work the problem has been formulated in *static* conditions, i.e., in a scenario where the aggregation is performed (relatively) shortly before the users' departure time. In other words, the state of the network can be assumed to be known and fixed. However, if accurate discrete-time measurements of the network state were available, thanks to the employment of suitably-designed widespread sensing technologies, and if such measurements were updated with a sampling period of amplitude δ , then the polylines x_{ij} and the associated quantities could be rewritten as

$x_{ij}[z]$	Polyline connecting node $v_i \in \eta$ to node $v_j \in \eta$ at time $t \in [z\delta, (z + 1)\delta[$, with $z \in \mathbb{Z}^+$.
$w_{ij}[z], l_{ij}[z], \tau_{ij}[z]$	Cost, length and travel time of x_{ij} at time $t \in [z\delta, (z + 1)\delta[$, with $z \in \mathbb{Z}^+$.

Hence, by considering the following time-varying decision variables

$$X_{ijkd}[z] = \begin{cases} 1 & \text{if the } k\text{-th polyline of the } d^{\text{th}} \text{ driver} \\ & \text{solution path coincides with } x_{ij}[z], \\ 0 & \text{otherwise,} \end{cases}$$

the solution procedure could be readily extended to the *dynamic* case, thus enabling also event-driven re-planning. In this respect, according to the BONVOYAGE platform architecture presented in [15][16], the authors are focused on ensuring (i) the real-time monitoring of the dynamic present context (accounting for the multi-modal transportation network dynamics by means of a suitable multi-layer graph [15][17]) and (ii) the exploitation of the user feedbacks

provided in real-time by the users themselves (e.g., the Quality of Experience level perceived by each user during a given travel), in order to enable re-planning any time the occurrence of an event in the considered transportation network seriously compromises the feasibility of the currently selected travel solution. Indeed, the exploitation of user feedbacks and the real-time monitoring of the dynamic present context enable the *closed-loop control* mechanism that, by relying on similar methods as those used in [18]-[35], represents one of the most innovative features of the BONVOYAGE platform. Finally, it is worth mentioning the fact there are several interpretations of w_{ij} , l_{ij} and τ_{ij} : for instance, as per-unit or absolute quantities.

Once Step (1) is solved, the formulation (1)-(16) allows to generate all optimal drivers' routes as solutions to the considered M2M carpooling problem. However, due to the huge dimensions of real-sized scenarios, in the following simulations we resort to a heuristic approach.

III. SIMULATIONS

To evaluate the effectiveness of our heuristic approach, in this section simulations on a real scenario of medium dimensions are shown. More in detail, a portion of the city of Bilbao, Spain, covering an area of 8.75 km² has been considered. The data needed for building the transport networks \mathcal{G}_{ped} , \mathcal{G}_{pub} and \mathcal{G}_{pri} have been downloaded from OpenStreetMap [36] and the multigraph $\mathcal{G} = \mathcal{G}_{\text{ped}} \cup \mathcal{G}_{\text{pub}} \cup \mathcal{G}_{\text{pri}}$ consists of 4981 nodes and 9790 edges.

A. Algorithm

As already mentioned, the problem complexity imposes the use of some heuristic procedure. The adopted approach is structured into two stages, namely the *Pre-Processing Phase* and the *Main Processing Phase*. The algorithmic structures of these two stages are described below.

- **Pre-Processing Phase.** The goal of this step is to compute a matrix S^* encoding the similarity degree between users (classified into *Passengers* and *Drivers*). For each passenger-driver couple, at first it is checked if hard constraints (whose satisfaction passengers cannot do without) match: if this is the case, then the s_{ij} entry of the similarity matrix is computed according to the function *similarity(Driver d_i , Passenger p_j)* which takes into account soft constraints, behavioural affinity and spatial proximity (for the detailed implementation, the reader is referred to [37]). The problem dimensions are reduced at this stage since, by checking whether hard constraints do match or not, the set of users is restricted to only those eligible for aggregation.

PRE-PROCESSING PHASE
Input: <i>Drivers, Passengers</i>
Output: Similarity Matrix S^*
1: <i>PassengersToAggregate</i> \leftarrow empty set
2: <i>DriversToAggregate</i> \leftarrow empty set
3: Similarity matrix $S \leftarrow$ zero matrix
4: for each driver $d_i \in \text{Drivers}$
5: for each passenger $p_j \in \text{Passengers}$
6: if hard constraints do match

```

7:    $S \leftarrow s_{ij} = \text{similarity}(d_i, p_j)$ 
8:   endif
9: endfor
10: endfor
11:  $S^* \leftarrow S$  removing all zero rows and columns
12:  $\text{PassengersToAggregate} \leftarrow$  columns of  $S^*$ 
13:  $\text{DriversToAggregate} \leftarrow$  rows of  $S^*$ 
return reduced similarity matrix  $S^*$ 

```

- **Main Processing Phase.** Here the very aggregation procedure is performed using the similarity matrix and the reduced set of users identified in the previous step. More in detail, this procedure is iterative and at each step an attempt is made to aggregate the passenger-driver couple exhibiting the highest similarity degree. At first, the shortest path for the current passenger is computed on the multigraph $\mathcal{G}_p^{(d)}$ through the function $SP(\text{Passenger}, \text{Graph})$. Then the driver's path for picking up and dropping off the current user is computed and, if for both users aggregation is convenient, the carpool associated with the current driver is updated. If in the carpool there are already other passengers, it is checked whether the modified path of the pool is acceptable; if not, the current user is discarded.

MAIN PROCESSING PHASE

Input: Similarity Matrix $S^* := [s_{ij}^*]$, $\text{DriversToAggregate}$, $\text{PassengersToAggregate}$

Output: carpools CP_{i^*}

```

1: while  $S^* \neq \emptyset$ 
2:   select driver-passenger couple  $(d_{i^*}, p_{j^*})$  such that
    $(i^*, j^*) = \arg \max_{i,j} [s_{ij}^*]$ 
3:   create a carpool  $CP_{i^*}$  associated with driver  $d_{i^*}$ 
4:   if the number of seats requested by  $p_{j^*}$  is less than or
   equal to those offered by  $d_{i^*}$ 
5:      $\text{path}_{p_{j^*}} \leftarrow SP(p_{j^*}, \mathcal{G}_p^{(d_{i^*})})$ 
6:     if  $\text{path}_{p_{j^*}}$  is more convenient than
        $SP(p_{j^*}, \mathcal{G}_{ped} \cup \mathcal{G}_{pub})$  and  $CP_{i^*}$  is empty
7:        $\text{newPath}_{d_{i^*}} \leftarrow$  shortest path for  $d_{i^*}$  passing
       through  $p_{j^*}$ 's entry and exit points
8:       if  $\text{newPath}_{d_{i^*}}$  satisfies the requirements of
       driver  $d_{i^*}$  in terms of waiting time and
       number of transit nodes
9:         add passenger  $j^*$  to carpool  $CP_{i^*}$ 
10:        set column  $j^*$  of  $S^*$  to zero
11:        if carpool  $CP_{i^*}$  is full
12:          set row  $i^*$  of  $S^*$  to zero
13:        endif
14:      else if  $\text{path}_{p_{j^*}}$  is more convenient than
         $SP(p_{j^*}, \mathcal{G}_{ped} \cup \mathcal{G}_{pub})$  and  $CP_{i^*}$  is not empty
15:         $\{\text{newPaths}\} \leftarrow$  set of shortest paths for  $d_{i^*}$ 
        passing through the entry and exit points of
         $p_{j^*}$  and of those passengers who are already
        in  $CP_{i^*}$ 
16:        while  $\{\text{newPaths}\} \neq \emptyset$ 
17:           $\text{newPath}_{d_{i^*}} \leftarrow$  the fastest path in
           $\{\text{newPaths}\}$  according to the driver's
          point of view
18:          if  $\text{newPath}_{d_{i^*}}$  satisfies the requirements
          of  $d_{i^*}$  in terms of waiting time and transit

```

```

nodes and the requirements of the
previously aggregated passengers
19:   add passenger  $j^*$  to carpool  $CP_{i^*}$ 
20:   set column  $j^*$  of  $S^*$  to zero
21:   if carpool  $CP_{i^*}$  is full
22:     set row  $i^*$  of  $S^*$  to zero
23:   endif
24:    $\{\text{newPaths}\} = \emptyset$ 
25:   else
26:     remove  $\text{newPath}_{d_{i^*}}$  from  $\{\text{newPaths}\}$ 
27:   endif
28: endwhile
29: endif
30: endif
31:  $s_{i^* j^*}^* \leftarrow 0$ 
32: endwhile
return all carpools  $CP_{i^*}$ 

```

It is worth stressing out the fact that there is no one-to-one correspondence between these two phases and the two steps mentioned in Section II. Instead, from the algorithmic implementation viewpoint, Step (1) and Step (2) collapse into the Main Processing Phase (points 5 through 7). In the best-case scenario, the GD algorithm (i.e., Step (1)) is invoked once for each passenger (instead of M times). In addition, the constrained SPP from the driver's point of view (i.e., Step (2)) is simplified due to the priority order assessed as a result of the Pre-Processing Phase.

B. Results

The simulations presented in this section have been carried out using MATLAB® on an Asus notebook, Intel Dual Core 2.90 GHz, 8GB RAM, running Windows 10.

First of all, a simple scenario with 2 drivers and 5 passengers is considered. The drivers are assumed to offer 3 seats each and to specify two distinct maximum detour ranges. Moreover, each passenger is assumed to require only one seat. Hence, the algorithm described in the previous subsection yields the creation of 2 carpools – each of them composed of a driver and 2 passengers – and only one passenger is left alone (i.e., passenger no. 1, because it is more convenient for him/her to use $\mathcal{G}_{ped} \cup \mathcal{G}_{pub}$). In particular, from Figs. 1-4 it is clear that the proposed algorithm solves an instance of the M2M carpooling problem, and therefore, *a fortiori*, can also be relied upon for solving instances of the MS-SD and SS-MD problems. Note that the polygon outlined by a black line and surrounding the planned path in Figs. 1 and 3 represents the area each driver is willing to sweep according to his/her detour range.

A second set of simulations has been carried out so as to assess the performances with respect to other solution approaches available in the literature. In this case, the map is populated with 35 passengers and 15 drivers. The number of seats offered by each driver ranges between 1 and 3 as well as the number of seats requested by each passenger. The drivers' detour range and the maximum number of transit nodes is changed at each simulation in order to reproduce different scenarios. Furthermore, the users' sources are chosen randomly, assuming that 70% of the users' destinations belong to a set of 3 nodes, representing attractive points which in an

urban context are likely to appear (e.g., schools, offices, shopping centers). Within this framework, the average number of aggregated passengers is 22 (i.e., 60% of the total number of passengers), while the number of empty carpools is zero (i.e., all drivers are aggregated with at least one passenger). The algorithm runtime ranges between 34 seconds and 2 minutes. This uncertainty in the algorithm runtime mainly depends on two factors: (i) the fact that a high similarity degree translates into a high aggregation probability, and (ii) the availability of other feasible travel solutions featuring carpooling, should the currently proposed one turn out to be infeasible for the driver.

Considering the best-case scenario, the proposed heuristics allow to drastically reduce the computational costs. More in detail, the algorithm runtime is one order of magnitude smaller than that reported in [12]. However, when making this comparison, it must be noted that the solutions proposed in the cited work, unlike ours, are designed on a day-ahead basis and thus, in such a context, fast computations are not mandatory. In [11], the authors report 2-3 seconds for solving the problem for each driver and a percentage of aggregated passengers around 70%. When comparing these last results with ours, it should be noted that (i) the map considered in [11] consists in a set of edges and vertices whose cardinality is one order of magnitude smaller than the cardinality of the edge and vertex sets considered in our simulations – the authors of [11] themselves claim that enlarging the map dimensions would lead to a considerable increase in terms of computational complexity – and (ii) the algorithm proposed in [11] is capable of bending the drivers’ trajectories only.

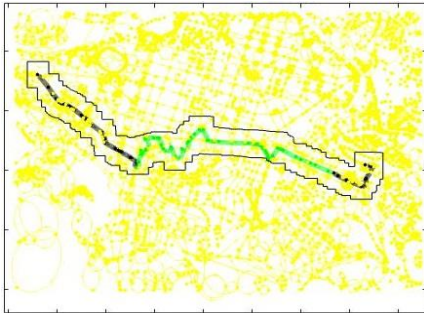


Figure 1. Planned path for driver no. 1 after aggregation with passengers nos. 4 and 5: in green the polyline representing the path of the first carpool, in black the polyline representing the solo part of the driver’s ride.

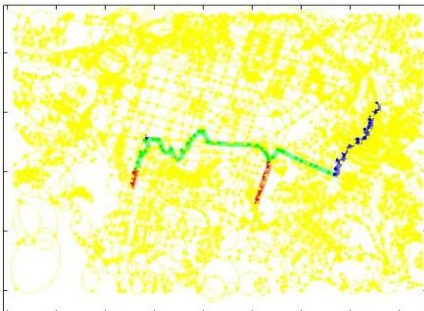


Figure 2. Planned path for passengers nos. 4 and 5 in the first carpool: in green the polyline representing the carpool path, in blue and red the polylines describing the solo paths of passengers nos. 4 and 5, respectively, before being picked up and after being dropped off.

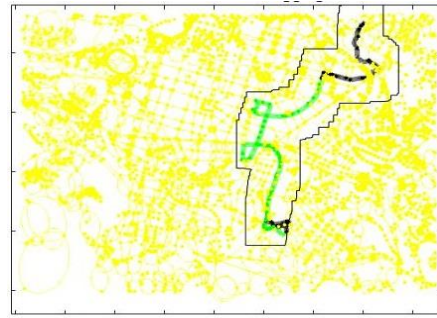


Figure 3. Planned path for driver no. 2 after aggregation with passengers nos.2 and 3: in green the polyline representing the path of the second carpool, in black the polyline representing the solo part of the driver’s ride.

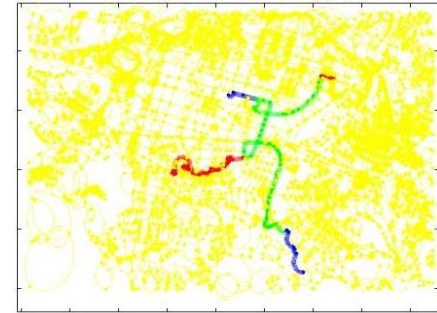


Figure 4. Planned path for passengers nos. 2 and 3 in the second carpool: in green the polyline representing the carpool path, in blue and red the polylines describing the solo paths of passengers nos. 2 and 3, respectively, before being picked up and after being dropped off.

IV. CONCLUSION

This work presents the mathematical formulation and the algorithmic implementation of the many-to-many carpooling problem with automated passenger aggregation in the context of multi-modal trip planning. The simulation results show that the proposed heuristic approach provides an efficient solution to the considered problem and its computation time is compatible with a real-time implementation. The proposed approach is truly multi-modal, in that it is capable of handling combinations of any private and public transportation modes (e.g., train, bus, electric car-sharing in urban areas [38][39]), ensuring that user preferences and constraints are simultaneously met. The authors plan to adjust the aggregation algorithm, for it to reach a high degree of scalability, thus exhibiting a reasonable computational complexity even as the multi-graph dimensions grow and the number of drivers and passengers is further increased.

ACKNOWLEDGMENT

The authors wish to thank the CRAT team (namely, Dr. S. Canale and Ing. F. Lisi), Prof. N. Blefari Melazzi and all the partners of the BONVOYAGE project for the constant and constructive cooperation and helpful suggestions.

REFERENCES

- [1] J. I. Levy, J. J. Buonocore, K. Von Stackelberg, “Evaluation of the public health impacts of traffic congestion: a health risk assessment,” *Environmental Health*, vol. 9, no. 65, 2010.
- [2] K. Zhang, S. Batterman, “Air pollution and health risks due to vehicle traffic,” *The Science of the total environment*, pp. 307-316, 2013.
- [3] European Environment Agency, *Air quality in Europe – 2016 report*, no. 28/2016.

- [4] T. Litman, "Smart Congestion Relief: Comprehensive Evaluation of Traffic Congestion Costs and Congestion Reduction Strategies," *Victoria Transport Policy Institute*, 2016.
- [5] J. T. Tuomisto, and M. Tainio, "An economic way of reducing health, environmental, and other pressures of urban traffic: a decision analysis on trip aggregation," *BMC Public Health*, vol. 5, no. 1, 2005.
- [6] I. B. Hartman, D. Keren, A. A. Dbai, E. Cohen, L. Knapen, A. U. Yasar, D. Janssens, "Theory and Practice in Large Carpooling Problems," *Procedia Computer Science*, vol. 32, pp. 339-347, 2014.
- [7] R. Baldacci, V. Maniezzo, A. Mingozzi, "An Exact Method for the Car Pooling Problem Based on Lagrangian Column Generation," *Operations Research*, vol. 52, no. 3, pp. 422-439, 2004.
- [8] S. Yan and C. Y. Chen, "A model and a solution algorithm for the car pooling problem with pre-matching information," *Comput. Ind. Eng.*, vol. 61, no. 3, pp. 512-524, 2011.
- [9] S. Di Martino, R. Galiero, C. Giorio, F. Sarro, "A Matching-Algorithm based on the Cloud and Positioning Systems to Improve Carpooling," *Proceedings of the 17th International Conference on Distributed Multimedia Systems*, 2011.
- [10] Y. Guo, G. Goncalves, and T. Hsu, "A Clustering Ant Colony Algorithm for the Long-Term Car Pooling Problem," *Int. J. Swarm. Intell. Res.*, vol. 3, no. 2, pp. 39-62, 2012.
- [11] T. A. Son, L. T. H. An, P. D. Tao, and D. Khadraoui, "A distributed algorithm solving multiobjective dynamic car pooling problem," *2012 International Conference on Computer & Information Science (ICIS)*, Kuala Lumpur, pp. 231-236, 2012.
- [12] J. Xia, K. M. Curtin, W. Li, Y. Zhao, "A New Model for a Carpool Matching Service," *PLoS ONE*, vol. 10, no. 6, 2015.
- [13] <http://bonvoyage2020.eu/>
- [14] S.S. Biswas, B. Alam, M.N. Deja, "Generalization of Dijkstra's Algorithm for Extraction of Shortest Paths in Directed Multigraphs," *Journal of Computer Science*, vol. 9, no. 3, pp. 377-382, 2013.
- [15] S. Canale, A. Di Giorgio, F. Lisi, M. Panfili, L. Ricciardi Celsi, V. Suraci, F. Delli Priscoli, "A Future Internet oriented user centric extended intelligent transportation system," in *Proc. of the 2016 24th Mediterranean Conference on Control and Automation (MED)*, Athens, 2016, pp. 1133-1139. DOI: 10.1109/MED.2016.7535967.
- [16] A. Palo, L. Zuccaro, A. Simeoni, V. Suraci, L. Musto, P. Garino, "A common open interface to programmatically control and supervise open networks in the Future Internet," *2013 Future Network and Mobile Summit, FutureNetworkSummit 2013*, ISBN: 978-190582437-3.
- [17] R. Gambuti, S. Canale, F. Facchinei, A. Lanna, A. Di Giorgio, "Electric vehicle trip planning integrating range constraints and charging facilities," in *Proc. of the 23th Mediterranean Conference on Control and Automation (MED 2015)*, Torremolinos, Spain, 16-19 June 2015, pp. 472-479. DOI: 10.1109/MED.2015.7158793.
- [18] F. Delli Priscoli, A. Isidori, "A Control-Engineering Approach to Integrated Congestion Control and Scheduling in Wireless Local Area Networks," *Control Engineering Practice*, IFAC (Great Britain), vol. 13, no. 5, May 2005, pp. 541-558.
- [19] C. Bruni, F. Delli Priscoli, G. Koch, I. Marchetti, "An Optimal Approach to the Connection Admission Control Problem," *International Journal of Control*, Elsevier Science Pub., vol. 79, no. 10, October 2006, 1237-1250.
- [20] F. Delli Priscoli, F. Di Paolo, A. Fiaschetti, A. Pietrabissa, "A Robust Adaptive Congestion Control for Communication Networks with Time-Varying Delays," *Proceedings of the IEEE Conference on Control Applications*, Munich (Germany), October 4-6, 2006, pp. 2093-2098.
- [21] F. Delli Priscoli, A. Isidori, L. Marconi, "A method for robust regulation of non-minimum-phase linear systems," *Proc. 6th IFAC Symposium on Robust Control Design, ROCOND'09*, Haifa; Israel; 16-18 June, 2009, pp. 255-260, DOI: 10.3182/20090616-3-IL-2002.0036.
- [22] V. Suraci, M. Castrucci, G. Oddi, A. Cimmino, R. Colella, "Convergence in Home Gigabit Networks: Implementation of the Inter-MAC Layer as a Pluggable Kernel Module," *IEEE PIMRC 2010 Services, Applications, and Business - 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, DOI: 10.1109/PIMRC.2010.5671761.
- [23] C. Bruni, F. Delli Priscoli, G. Koch, I. Marchetti, "Resource Management in Network Dynamics: an Optimal Approach to the Admission Control Problem", *Computers & Mathematics with Applications*, Elsevier, Great Britain, Vol. 59, 2010, pp. 305-318.
- [24] G. Oddi, D. Macone, A. Pietrabissa and F. Liberati, "A proactive link-failure resilient routing protocol for MANETs based on reinforcement learning," *2012 20th Mediterranean Conference on Control & Automation (MED)*, Barcelona, 2012, pp. 1259-1264, doi: 10.1109/MED.2012.6265812.
- [25] S. Canale, F. Delli Priscoli, A. Di Giorgio, A. Lanna, A. Mercurio, M. Panfili, V. Suraci, "Resilient Planning of PowerLine Communications Networks Over Medium Voltage Distribution Grids," *20th Mediterranean Conference on Control and Automation (MED12)*, Barcelona (Spain), July 2012, pp. 710-715.
- [26] F. Delli Priscoli, M. Iannone, A. Pietrabissa, V. Suraci, "Modelling Quality of Experience in Future Internet Networks," *Future Network & Mobile Summit 2012*, Berlin, July 2012.
- [27] C. Bruni, F. Delli Priscoli, G. Koch, A. Pietrabissa, L. Pimpinella, "Network Decomposition and Optimal Multipath Routing Control Problem for Load Balancing," *Transactions on Emerging Telecommunications Technologies (ETT)*, John Wiley & Sons, Inc., USA, vol. 24, no. 2, March 2013, pp. 154-165.
- [28] G. Oddi, A. Pietrabissa, F. Delli Priscoli, V. Suraci, "A decentralized load balancing algorithm for heterogeneous wireless access networks," in *Proceedings of the WTC 2014 (World Telecommunications Congress 2014)*, pp. 1-6, ISBN: 978-380073602-7.
- [29] L. Ricciardi Celsi, S. Battilotti, F. Cimorelli, C. Gori Giorgi, S. Monaco, M. Panfili, V. Suraci, and F. Delli Priscoli, "A Q-Learning Based Approach to Quality of Experience Control in Cognitive Future Internet Networks," in *Proc. of the 23rd Mediterranean Conference on Control and Automation (MED15)*, pp. 1045-1052, June 16-19, 2015, Torremolinos, Spain, DOI: 10.1109/MED.2015.7158895.
- [30] L. Ricciardi Celsi, R. Bonghi, S. Monaco, and D. Normand-Cyrot, "On the Exact Steering of Finite Sampled Nonlinear Dynamics with Input Delays," in *Proceedings of the 1st Conference on Modelling, Identification and Control of Nonlinear Systems (MICNON 2015), IFAC-PapersOnLine*, vol. 48, no. 11, pp. 674-679, June 24-26, 2015, Saint-Petersburg, Russia.
- [31] S. Battilotti, C. Gori Giorgi, S. Monaco, M. Panfili, A. Pietrabissa, L. Ricciardi Celsi, and V. Suraci, "A Multi-Agent Reinforcement Learning Based Approach to Quality of Experience Control in Future Internet Networks," in *Proc. of the 34th Chinese Control Conference (CCC2015)*, pp. 6495-6500, July 28-30, 2015, Hangzhou, China, DOI: 10.1109/ChiCC.2015.7260662.
- [32] C. Bruni, F. Delli Priscoli, G. Koch, A. Palo, A. Pietrabissa, "Quality of Experience Provision in the Future Internet," *IEEE Systems Journal*, vol. 10, no. 1, March 2016, pp. 302-312.
- [33] F. Cimorelli, F. Delli Priscoli, A. Pietrabissa, L. Ricciardi Celsi, V. Suraci, and L. Zuccaro, "A Distributed Load Balancing Algorithm for the Control Plane in Software Defined Networking," in *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED 2016)*, pp. 1033-1040, June 21-24, 2016, Athens, Greece, DOI: 10.1109/MED.2016.7535946.
- [34] A. Pietrabissa, "Distributed stochastic multi-vehicle routing in the Euclidean plane with no communications," *International Journal of Control*, vol. 89, no. 8, 2016, pp. 1664-1674, doi: 10.1080/00207179.2016.1144237.
- [35] F. Delli Priscoli, A. Di Giorgio, F. Lisi, S. Monaco, A. Pietrabissa, L. Ricciardi Celsi, V. Suraci, "Multi-Agent Quality of Experience Control," *International Journal of Control, Automation, and Systems*, vol. 15, no. 2, pp. 892-904, 2017, DOI: 10.1007/s12555-015-0465-5.
- [36] <https://www.openstreetmap.org/>
- [37] <http://bonvoyage2020.eu/results/deliverables/>, D4.1, *Design of the Intelligent Transport Functionality*, chapter 5, H2020 BONVOYAGE project.
- [38] A. Di Giorgio, F. Liberati, R. Germanà, M. Presciuttini, L. Ricciardi Celsi, and F. Delli Priscoli, "On the Control of Energy Storage Systems for Electric Vehicles Fast Charging in Service Areas," in *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED 2016)*, pp. 955-960, June 21-24, 2016, Athens, Greece, DOI: 10.1109/MED.2016.7535947.
- [39] Di Giorgio A., Liberati F., Canale S., "Optimal electric vehicles to grid power control for active demand services in distribution grids," *20th Mediterranean Conference on Control and Automation MED12*, IEEE, 1309-1315, Barcelona, July 2012, DOI: 10.1109/MED.2012.6265820.