

Available online at www.sciencedirect.com**ScienceDirect**

Transportation Research Procedia 21 (2017) 110–129

**Transportation
Research
Procedia**

www.elsevier.com/locate/procedia

2016 International Symposium of Transport Simulation (ISTS'16 Conference), June 23~25, 2016

Analysis and Operational Challenges of Dynamic Ride Sharing Demand Responsive Transportation Models

MP. Linares, J. Barceló*, C. Carmona and L. Montero

Universitat Politècnica de Catalunya-Barcelona Tech UPC – Carrer Jordi Girona 1-3 08034 Barcelona

Abstract

There is a wide evidence that sustainable mobility is not only a technological question, automotive technology will be part of the solution as a necessary but not sufficient condition, sufficiency is emerging as a combination of a paradigm shift from car ownership to vehicle usage consequence of socio-economic changes, with the application of Information and Communication Technologies (ICT) that make possible for a user to have access to a mobility service from anywhere to anywhere at any time. Among the many emergent mobility services Multiple Passenger Ridesharing and its variants look the more promising. However, implementations of these systems accounting specifically for time dependencies, and time windows reflecting users' needs raise challenges in terms of real-time fleet dispatching and dynamic route calculation. On the other hand the feasibility and impacts analysis in terms of the many factors influencing the behavior of the system, as for example the service demand, the size of the service fleet, the capacity of the shared vehicles, the time windows requirements, soft or tight. This paper analyzes both aspects. The first is approached in terms of a Decision Support System whose solutions are computed in terms of ad hoc heuristics of variants of Pick Up and Delivery Problems with Time Windows and Feasibility and Profitability criteria rooted on Dynamic Insertion Heuristics. For the evaluation of the applications a Simulation Framework is proposed based on a microscopic simulation model that emulates real-time traffic conditions and a real traffic information system, and interacts with the Decision Support System feeding it with the required data to make the decisions that are implemented in the simulation to emulate the behavior of the shared fleet. The proposed simulation framework has been implemented in a model of Barcelona's Central Business District. The paper is completed with the discussion of the achieved results.

Copyright © 2017 The Authors. Published by Elsevier B.V.

Selection and Peer-review under responsibility of Dept. of Transportation Engineering, University of Seoul.

Keywords: Demand Responsive Transport; Pickup and Delivery with Time Windows; Dynamic Fleet Management; Microscopic Traffic Simulation

* Corresponding author. Tel.: +34 93 401 70 33; fax: +34 93401 58 55.

E-mail address: jaume.barcelo@upc.edu

1. Introduction

Urban areas must address from a holistic perspective the challenges and threads of sustainability namely in providing services to companies and citizens. Cities are complex systems, any city must be thought as a “System of Systems”, and Mobility is only one of the components of such complex systems, a non-isolated component strongly interacting with all other components and therefore its implications must be analyzed in the context of these interactions. Frost & Sullivan, in a recent analysis of the Future of Mobility and New Mobility Business Models, (Sullivan & Frost 2015) identify the growing trend of “Ride Sharing” models as one of what they call “Transformational Shifts in Mobility”. This trend can be seen as one of the consequences of the paradigmatic shift from “car ownership” to “vehicle usage”, leading to a new concept of multi modal mobility network, as a way of overcoming the limitations of conventional public transport systems, made possible by the pervasive penetration of Information and Communication Technologies (ICT). Technology enables a comfortable seamless real-time point to point travel service.

Demand Responsive Transport (DRT), Dial-a-Ride Transit or Flexible Transport Services, according with the definition of the European Commission, Directorate-General for Energy and Transport “are emerging user-oriented forms of public transport characterized by flexible routing and scheduling of small/medium vehicles operating in shared-ride mode between pick-up and drop-off locations according to passenger needs”. DRT were initially thought to provide public transport services for areas with low passenger demand where regular bus services would not be available. However, this concept is quickly evolving enabled by the ICT deployment, initiatives like KUTSUPUS (Anon 2015), an on-demand minibus service run by Helsinki’s public transit authority, letting riders choose their own route summon a trip with a smartphone, decide the start and end point of their trip and choose whether to share a journey or not. It is a new Demand Responsive Public Transport service designed to achieve maximum flexibility.

This concept of Demand Responsive Transport is rapidly evolving to services provided by private companies operating point to point with full dynamics and flexibility, also offering the possibility of sharing trips. E-hailing is a process of ordering a transportation service by a private car (e.g. Uber services), special taxi services, etc. The system has currently a variety of implementations but, in essence, the variant we are interested in this paper, assumes that the customer books or hails the trip electronically providing the pickup location (that can be automatically identified by GPS current customer location), the drop off location, and the desired pickup up and drop off time windows, and that multiple passengers can share the trip. In order words our research addresses specifically the variant known a “Multiple Passengers Ridesharing System”.

A state-of-the art-survey on the variants of ridesharing systems, their alternatives and likely future evolution can be found in (Furuhata et al. 2013), according to their classification the variant studied in this paper corresponds to the called Detour Ridesharing with Multiple Passengers, in which multiple passengers, with relatively close but different origins and destinations share rides which can partial or totally overlap. Our research has analyzed the potential utilization uses of special fleets of dedicated vehicles in an urban area, assuming than further than the pickup and drop off location and time windows, the system is also aware of the current and desirably short term forecasted traffic conditions to timely determine the optimal routes satisfying customers’ time constraints. That is we assume that the system is operating in a network for which an Advanced Traveler Information System (ATIS) provide the travel time estimates. This type of real-time ridesharing system has also been studied by (Ma et al. 2015) but with simplifications concerning the availability of traffic information. The special case when the fleet of service vehicles is composed of autonomous vehicles has deserved special attention from the agent-based simulation approach, a general perspective can be found in (Fagnant & Kockelman 2014) that analyze the environmental impacts, while (Martinez et al. 2015) propose a general agent-based simulation to assess the impacts, and apply it to the city of Lisbon. This work has been the basis for the report of the International Transport Forum (OECD - International Transport Forum 2015). One of critical aspects highlighted in these last references concerns the operational efficiency of the system, determined, among other factors by the fleet management dispatching system, in terms of the decisions process and its dependencies with the size of the fleet and the demand, an aspect that has been studied in (Boesch et al. 2016) for a simplified dispatching strategy. Consequently our work has dealt namely with the Decision Support System supporting the decision making process of which vehicle assign to optimally

pride the requested service for a customer, taking into account the new service requirements, the confirmed rights of the customers already being provided a service, and the current and estimated travel times in the urban network. The decision support system, inspired on a previous work (Barceló et al. 2013) for pickup and delivery real-time services with time windows (PDPTW), develops an ad hoc version of PDPTW with time-dependent shortest paths assuming that time dependencies are provide by an ATIS.

The proposed system has been tested by microscopic traffic simulation using a model of the Barcelona’s Business District, defining specialized fleets and using as design factors in the simulation experiments the demand of shared vehicles, the sizes of the shared vehicles, the size of the fleet providing the service, the conditions on the customers’ time windows (e.g. tight or soft) and the degree of dynamism, that is the % with respect to the total demand of the ride-sharing services that is not known before the start of the journey. Computational results are presented and discussed.

2. Description of the proposed cooperative urban mobility system

The proposed Real-Time Multiple Passenger Ridesharing system object of this paper, and how it works, is depicted in the simple example of Fig. 1. Let’s assume a fleet consisting of two vehicles V1 and V2, both with initially assigned plans which have been previously optimized. At a given time a customer C1 calls the system for a service, the customer’s location is automatically recorded by the system. The customer tells the system which is his/her time expectancy of pick-up, in other words the time window (e_1, l_1) of his/her waiting time expectancy (e.g. e_1 could be the time at which the call is made, and l_1 the latest time he/she expects to be pick up). The customer also informs the system of his/her destination that is delivery point, D1 and likely, the time at which he/she would approximately like to arrive at the destination, let’s say ($a_1 \pm \epsilon_1$), where ϵ_1 represents an acceptable slack time.

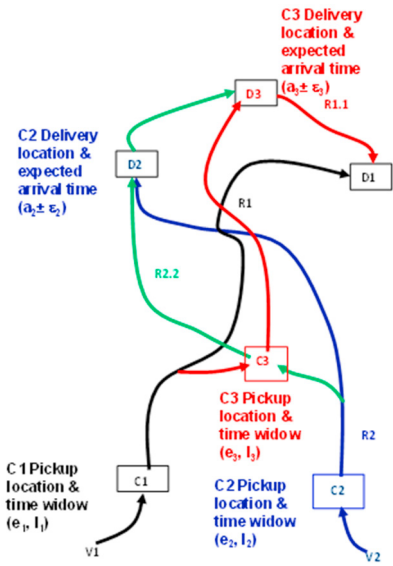


Fig. 1. Example of the real-time Multiple Passengers Ridesharing System.

The system, which is aware of locations and status of each vehicle in the fleet (e.g. fleet vehicle are GPS tracked, and the ICT functions inform the system on the current level of occupancy of the vehicle, that is the number of passengers, their destinations and time constraints), as well as the current network conditions (e.g. a real-time traffic information system keeps the Decision Support System updated about travel times, congestions, incidents, and so on), determines which of the vehicles is the most appropriate to provide the service to customer C1 both in terms of quality of the service and profitability. Let’s assume that on basis to the available information the Dispatching System assigns the service to vehicle V1 and the route R1 to pick up the customer and take it to its destination.

In a similar way, let’s assume that customer C2, who is located at C2, has to be pick up within the time window

(e_2, l_2) and wants travel to destination D_2 , where expects to arrive around ($a_2 \pm \epsilon_2$), is assigned to vehicle V_2 that will follow route R_2 to pick him/her up and travel to the destination D_2 .

The critical situation showing how the performance of the systems depends on the quality of the Decision Support System that manages the fleet, and dispatches the services, arises when some time later, let's say at time t , a new customer C_3 , at location C_3 , asks for a service to pick up him/her within the time window (e_3, l_3) to travel to destination D_3 where he/she expects to arrive around ($a_3 \pm \epsilon_3$).

In order to provide the new customer the requested service, the system faces many alternatives. First, it can open a new route assigning one of the empty vehicles of the fleet to the new client, but this action may imply a high cost. A better alternative would be to assign one of the en-route vehicles that are closer to the client, if the time constraints of the customers that are already being served allow this. In this example, both proposed solutions accept a diversion policy, so, the vehicles could divert from its original route to serve the new client and return to its assigned scheduled.

One of the alternatives is to assign the new customer service to the vehicle V_1 , which exchanges its route R_1 for the new route $R_{1.1}$ (described in the figure with red arrows). This vehicle picks up customer C_3 within the accepted time windows, and then it drives to the D_3 destination to deliver the customer. After this, vehicle V_1 follows its new route and delivers customer C_1 at the corresponding destination D_1 taking into account the time constraints of all the served customers. The second possibility is that the new service of customer C_3 is assigned to vehicle V_2 , which diverts its route to pick up the customer by following route $R_{2.2}$ (described in the figure with green arrows). In this case, the vehicle before delivers customer C_2 in its destination D_2 and after, it goes to the destination D_3 of customer C_3 . Also in this case, we need take into account the time constraints with respect to all customers involved in. The decision will be made accounting for criteria that while ensuring the quality of the service provided to the customers achieves the maximum system profitability.

3. General framework

The feasibility and performance of the proposed Multiple Passenger Ridesharing system has been tested by simulation using the conceptual framework depicted in Fig. 2.

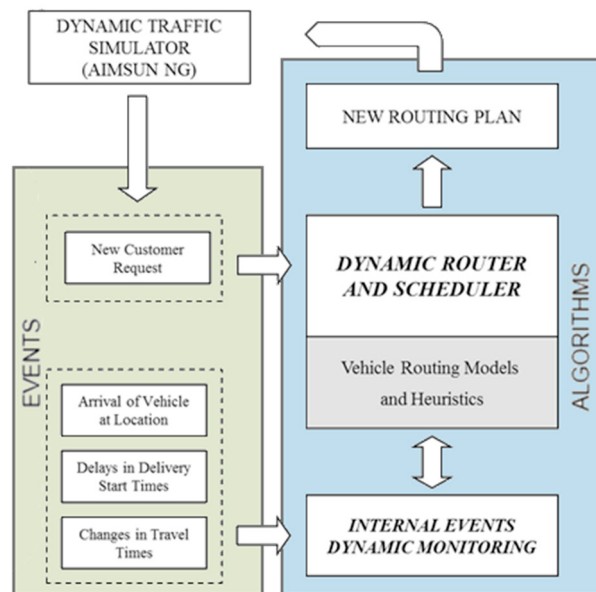


Fig. 2. General conceptual framework.

The framework is based on the combination of dynamic traffic simulation and the algorithms defined on the Dynamic Routed and Scheduler (DRS). The dynamic traffic simulator offers the possibility to emulate with great

detail and realism the traffic conditions on a road network, as well as the information provided by ICT (for instance, by GPS). Taking advantage of this capability, the simulator supplies all information needed during the process, and performs the other tasks required along the global process. The DRS is the core of the framework and consists on a set of algorithms providing on-line solution to the real-time vehicle routing problem with time windows and time dependencies.

The DRS is triggered every time an event is detected by the system. We consider two types of events: external and internal. In this particular case, the external event depends on the demand. Internal events, in contrast, depend on the observed traffic flow dynamics which could be the cause of delays in the estimated arrival times to the location of customers. So, the inputs of the DRS are the different types of events, with their corresponding attributes, that randomly occur during the fleet operations. In this work, we use a dynamic traffic simulator that generates these events according to realistic stochastic patterns while tracking the position of vehicles. The internal events dynamic monitoring continuously checks the traffic conditions on the network as well as the fleet expected timing. When a new reassignment takes place, a new operational plan is generated and returned to the dynamic simulator. The process continues until the end of the evaluation period.

3.1. *Dynamic Router and Scheduler*

As core of the computational framework the Dynamic Router and Scheduler consists on a set of algorithmic tools providing on-line solutions to the proposed real-time vehicle routing problems. The algorithm used can be exact algorithms, heuristics or hybrid approaches.

In the case analyzed in this paper we focus on the use of dynamic insertion heuristic solutions based on feasibility and profitability concepts. Feasibility is considered in the sense that vehicles serving those routes have enough time to travel to the new customer taking into account the time windows constraints of all customers already being served. The profit of a new insertion is the negative of the additional travel time incurred from inserting the new customer in a route. These algorithms are described in detail in the next Section.

3.2. *Dynamic Traffic Simulator*

The second component of the computational framework is a dynamic traffic simulator, which is used as a tool that:

- Models the road network.
- Generates databases of time-dependent link travel times.
- Randomly generates the different external and internal events previously mentioned.
- Moves vehicles along the road network according to a defined plan, keeping track of their positions and state at every time step of the simulation.
- Guides vehicles through the network using the current shortest paths information.
- Provides to the Dynamic Router and Scheduler the input data required by the algorithms

In this work we selected Aimsun v7 (Transport Simulation Systems 2013) among the currently existing microscopic traffic simulators because its functional architecture supports various extending modeling utilities, each one with a different role and objectives for working with external applications, as required to include the user algorithms implemented for the study. One of these possibilities is Micro API (APPI).

The Aimsun Micro API (application program interface) is a module that gives the simulator the ability to interface with almost any external application that may need access to some objects of the microsimulator during simulation run time as for example the Dynamic Router and Scheduler explained above. The exchange of information between the external application and the microsimulator is done in every simulation step, which is the time interval at which the estate of the simulation model is updated. The development language has been C++ and Python.

4. Dynamic Router and Scheduler

This section describes in detail the algorithm developed to solve the proposed vehicle routing problem taking into account the different types of demand requests considered in our research:

- A pre-booked demand for the beginning of the considered journey.
- A pre-booked demand for other time of the considered journey.
- An on-time demand that it must to be served at the same time that it is required.

We summarize explicitly the characteristics of each part of the global algorithm: a first phase that builds an initial routing plan only considering the static pre-booked demand for the beginning of the journey; and next an insertion heuristic based on the profitability concept that solves the situation when a new request is reported by a customer (on-time demand or pre-booked demand but not for the beginning of the journey).

4.1. Initial solution building phase

The methodology presented in this work needs of the specification of an initial solution in order to carry out the optimization heuristic applied in the following phases. So, it is in this first step, where the initial routing plan is computed considering the pre-booked demand that must be served at the beginning of the journey. To get right, the “beginning” must be determined empirically depending on the considered scenario.

First of all, in order to approximate the best requests assignment to the vehicle fleet, a pre-assignment phase is performed. The objective of this process is to calculate the best vehicle option to assign certain demand request taking into account what would happen if this demand cannot be assigned to its best vehicle option but to the second one. This process is of special importance due to the fact that the order in which the demand requests are introduced into the initial routes, significantly influences the quality of the solution obtained at the end of the process.

This procedure assigns all the fleet vehicles to each demand request taking into account the cost of directly assign this demand to each of the vehicles. After that, this procedure calculates the differences among serving the demand from the best vehicle option, or from the second best, and so on. These values are called savings. This information is useful to perform the assignment of the demand into the vehicles. The idea, based on the parallel insertion method by (Christo-fides et al. 1979), is to insert first the demand request whose cost would be worst if it is not assigned to its best vehicle option (the greater saving).

After that, and taking into account the information achieved in the first phase, the demand requests are definitely assigned to its best vehicle option considering the feasibility constraints of the built routes. This second step of the initial solution constructing approach it is performed on the ground of the Simple Insertion Heuristic (Solomon 1987), according to which, the pickup and delivery pairs are introduced in a route in such a way that the increment of its cost is minimal and the considered constraints regarding: vehicles capacity, time windows, pairing and precedence rules are respected.

Finally, inter-route post-optimization methods are introduced in order to improve the previously calculated initial routing plan. The candidate routines include two inter-route improvement operators: the Shift Operator and the Exchange Operator. The aim is to find such modifications of the collection of routes previously obtained, that the newly created result is feasible and characterized by a smaller cost.

Initial Solution Building Phase

Load the pre-booked demand
 Pre-assignment phase Insertion
 Heuristic
 Post-optimization methods
 Shift Operator
 Exchange operator

4.2. Alternative Dynamic Single Pair Insertion (ADSPI)

The proposed heuristic is an adaptation of the Dynamic Single Pair Insertion (SPI) procedure used by (Grzybowska & Barceló 2012) to solve the dynamic version of the PDVRPTW (Pick and Delivery Vehicle Routing Problem with Time Windows). This procedure is in turn an adaptation of the Dynamic Insertion Heuristic (DINS), a greedy heuristic derived from the basic insertion heuristic proposed in (Campbell & Savelsbergh 2004). In this case, every un-routed customer is evaluated at every insertion point. The evaluation of this movement consists in checking both feasibility and profitability of every insertion; (Orozco & Barceló 2012) adapted DINS for the case of the dynamic VRPTW (Vehicle Routing Problem with Time Windows) by introducing some new elements to reflect the dynamics of the operations of a fleet in an urban environment.

Similarly as DINS, in (Grzybowska & Barceló 2012) SPI creates a new routing plan reckoning with dynamic factors affecting the addressed problem. Hence, in contrast to the static approach, the travel, arrival and waiting time for all un-served customers are calculated with respect to the triggering instants and depend on the moment a trip between two customer starts. Among all the candidates routes (in which it is feasible to introduce the new pair of customers) there is selected the one, which characterizes with the smallest value of the increment of the total cost.

In this paper, the objective of the proposed procedure is to incorporate a newly reported request (pickup and delivery stops) in the most favorable location in one of the existing built routes without violating any of the established constraints, respecting: capacity of the vehicles, time windows of the demand request, pairing and precedence. In addition, we also consider time-dependent shortest paths assuming that time dependencies are provided by an ATIS.

We assume that the initial routing plan has been computed through the above explained building phase methodology. The general idea of the algorithm is as follows: when a new demand request must to be processed, the algorithm checks the current state of the vehicles (load, position, route, assigned demand requests); then, it is incorporated into the existing routes or, if necessary, used to create a new route corresponding to a vehicle that is parked waiting to new demand.

Specifically, taking into account the demand request previously assigned to each route vehicle, the algorithm checks the feasibility related with the insertion (of this demand request into each of the vehicles of the fleet) with respect to the capacity constraints. The unfeasible routes are rejected. Then, the algorithm checks the feasibility with respect to the time windows constraints. Then, routes with insufficient time to serve the new demand request within their schedule are rejected. Finally, the algorithm selects, among the candidate vehicle routes, the route with the minimum added cost with respect to the cost of the original route (before the new insertion). We assume that requests are attended in a one-by-one basis.

4.2.1. Algorithm

The new demand request (pickup and delivery stops pair) is inserted into the existing routed fulfilling the: pairing, precedence, time windows and capacity constraints. The first two constraints are enforced during the insertion process. Thus, the feasibility check regards verification if both the time windows and the capacity constraints are violated.

Feasibility

In order to evaluate the feasibility of the insertion of a demand request in certain pair position (i,j) in a certain route, we must verify if either the vehicle's capacity or demand request's time window constraints are violated.

So, certain route keeps feasible after the insertion of the demand request if the pickup insertion between stops i and $i+1$ and the delivery insertion between stops j and $j+1$ satisfy:

- **Capacity Feasibility:** The idea is to check if with this new insertion the capacity constraint is or is not violated. We need to calculate the capacity of the vehicle at each pickup or delivery affected stops. The crucial point would be the pickup stops, where we need always to arrive with the needed space to serve the demand of that pickup location stop.
- **Time Windows Feasibility:** For all the pickup or delivery stops in this route, the forecasted arrival time of each of the "affected" stops (after the corresponding update) is smaller than its pickup/delivery time window upper bound (last available time).

Profitability

The profit of an insertion is defined as the negative of the additional cost (travel time) incurred from inserting the new customer request (pickup and delivery stops) in a vehicle route (Barceló et al. 2013). In particular, the procedure calculates the cost of the insertion of the pickup stop after the position i of the route and the insertion of the delivery stop after the position j of the route.

It is important to note that this insertion could change the cost of the route. But, not only the additional cost of the insertion, but also the cost for arriving to the pickup or delivery stops that are located after i into the route. This is because the travel cost depends on the departure time, and the departure time changes if a new demand is inserted into the route.

5. System Architecture

In order to implement the proposed framework to test the feasibility and performance of the Multiple Passenger Ridesharing system, we implement a complex system architecture. Fig. 3 shows this designed architecture.

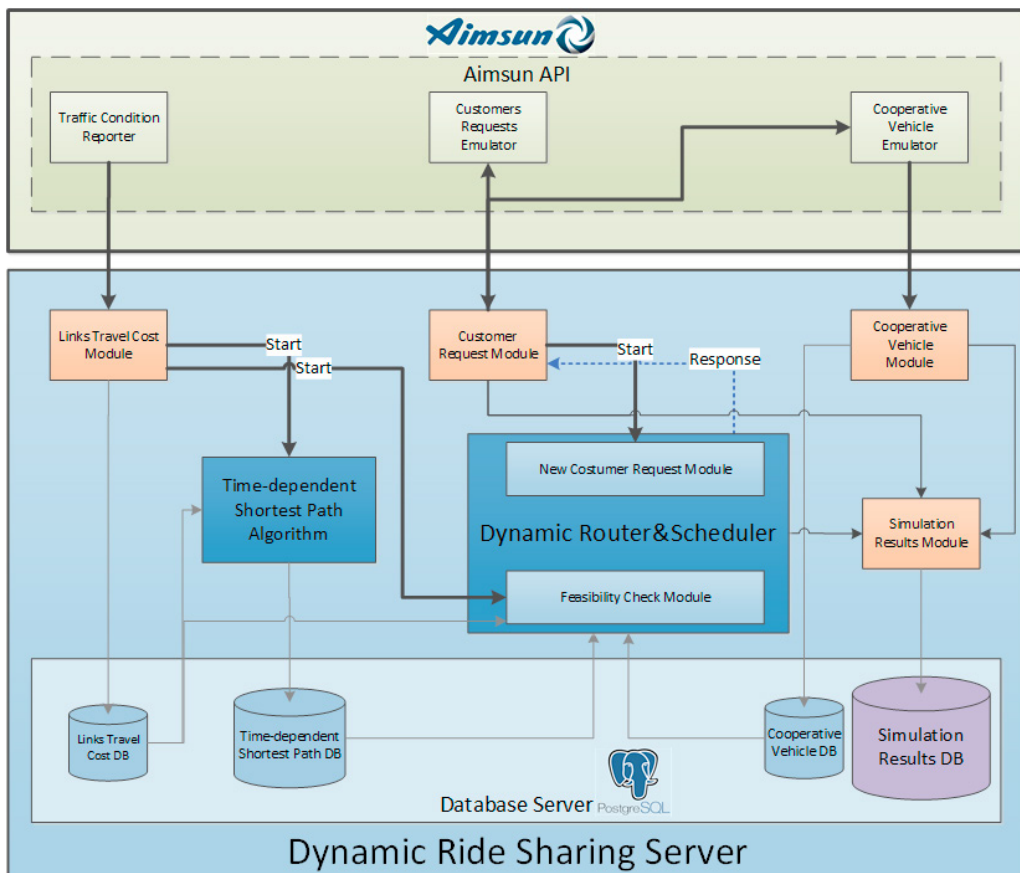


Fig. 3. System's Architecture.

5.1. Simulator API

In the proposed architecture the simulator API is used to develop three different emulators which exchange information with the Dynamic Ride Sharing Server.

- **Traffic Condition Reporter**
To send to the Dynamic Ride Sharing Server the traffic conditions (link travel cost for every link on the network) every given interval of time.
- **Customers' Request Emulator**
To emulate the demand requests. It allows the generation of the pre-booked demand and the dynamic requests.
- **Fleet Vehicle Emulator**
To provide the emulation of the vehicle fleet. It creates the vehicle fleet at the beginning of each simulation, sends to the Dynamic Ride Sharing Server the position (UTM coordinates) and state (customer pickup and destination stops) of every vehicle every given interval of time, and it makes every fleet vehicle to follow its assigned route.

The communication with the Dynamic Ride Sharing Server is performed implementing TCP sockets, with a TCP server listening in a port, one for each Aimsun API. A specific protocol has been designed for each of the different kind of data provided by the APIs.

5.2. Dynamic Ride Sharing Server

The Dynamic Ride Sharing Server hosts the different modules needed to provide an on-line solution to the real-time vehicle routing problem.

- **Link Travel Cost Module**
To receive the messages sent by the Traffic Condition Reporter. Save the link travel costs into the Link Travel Cost DB, and starts an execution of the Time-dependent Shortest Path Module in order to update the Time-dependent shortest path costs. Once the execution of the time-dependent shortest path module has ended this module starts an execution of the Feasibility Check Module, belonging to the DRS Module.
- **Customer Request Module**
To receive the information provided by the Customer Requests Emulator. Each petition will be a new customer order, and will fire a New Customer Request Module execution, inside the DRS Module. The information concerning the customer request will be saved by the KPI Module in order to calculate the proposed Key Performance Indicators (KPI).
- **Fleet Vehicle Module**
To receive the messages sent by the Fleet Vehicle Emulator. The information received will be saved into the Cooperative Vehicle DB, maintaining updated information of the vehicle fleet state.
- **Time-dependent Shortest Path Module**
It is the responsible of the time-dependent shortest path calculation. All the routes and its travel times from any section to any section of the network will be calculated and stored into the *Time-dependent Shortest Path DB*. Its required inputs are:
 - The link travel costs provided by the *Traffic Condition Reporter*, stored into the *Links Travel Cost DB*.
 - The *network graph*. This graph, stored in a file into the server, will be generated offline during the installation phase.
- **Dynamic Router and Scheduler Module**
It contains two modules in order to respond to two different problems:
 - To give an answer to new customers requests, assigning a cooperative vehicle to pick-up this customer and a route to reach the customer's destination.
 - To check if the previously assigned routes are feasible after a traffic condition update, recalculating the routes when needed. Feasibility is considered in the sense that vehicles serving those routes have enough time to travel to its passenger's destinations.

Its required inputs are:

- The time-dependent shortest path, stored into the *Time-dependent Shortest PathDB*.
- The cooperative vehicle fleet state reported by the *Fleet Vehicle Emulator* and stored into the *Fleet Vehicle DB*.
- The *network graph*. This graph, stored in a file into the server, will be generated offline during the installation phase.

It contains two different sub-modules:

- New Customer Request Module: when a new customer request arrives this module fires an execution, and the vehicle routing problem is solved. When a solution is reached, this module generates a response divided in two parts. On one hand, it informs the customer accepting its requests and giving him information about the vehicle assigned. On the other hand, the assigned vehicle receives a new route in order to pick up its new passenger.
- Feasibility Check Module: to update links travel costs in order to evaluate the current cost of the previously assigned routes. With all these inputs, this module checks if all the currently assigned routes are still feasible with the new traffic condition. If the current solution is not feasible, new routes will be calculated and cooperative vehicles affected by the routes changes will be informed with the new routes.

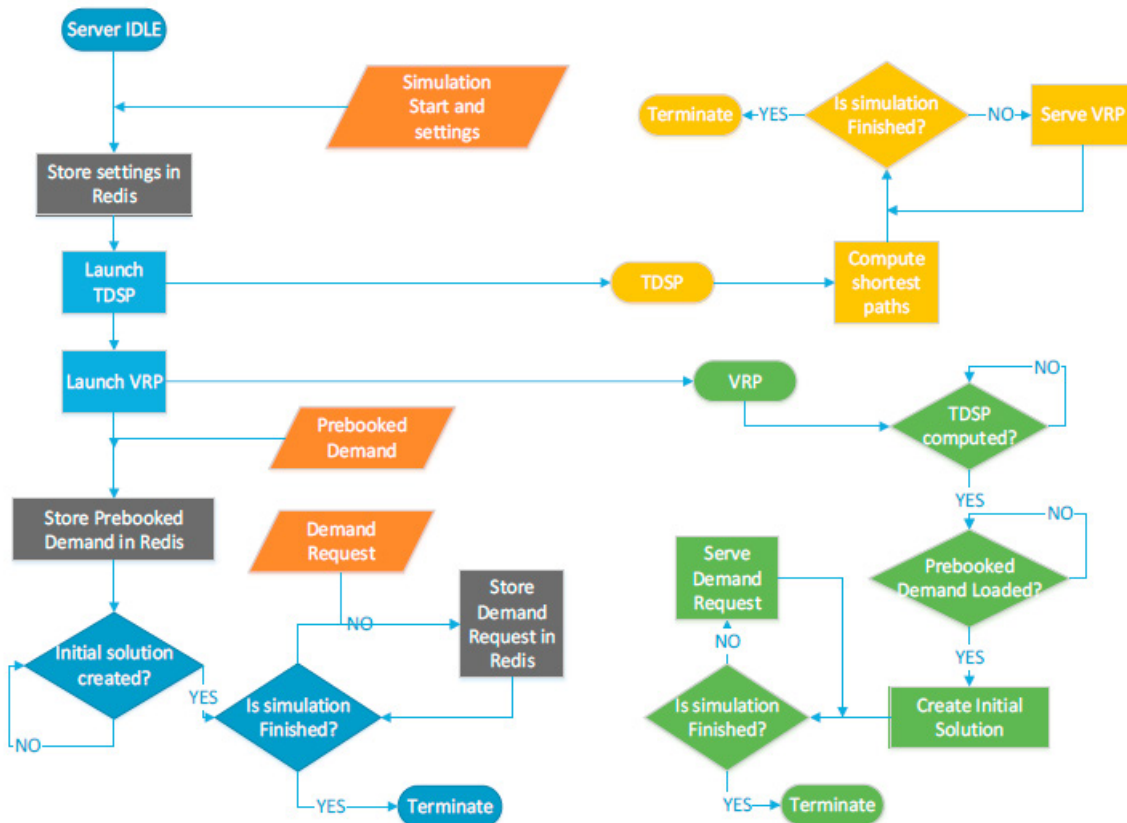


Fig. 4. Flow Diagram of the system process.

5.3. Database Server

A set of databases are needed to store both the information provided by the simulator as the information generated by the modules of the Dynamic Ride Sharing Server previously described.

- **Link Travel Cost Database**
To store the link travel cost provided by the Traffic Condition Reporter.
- **Time-dependent Shortest Path Database**
To store all the generated routes with its travel time from any section to any section of the network.
- **Fleet Vehicle Database**
To store the position (UTM coordinates) and state (customer pickup, arrival to destination) of every fleet vehicle.
- **KPI Database**
To store the values of the Key Performance Indicators calculated in every performed simulation.

Fig. 4 shows the flow chart that summarizes the described process.

6. Time-dependent Shortest Path Module

This module calculates the optimal vehicle paths among the pickup and delivery stops and the corresponding travel time. The shortest path problem considered in our research differs from the usual shortest path approaches in that we don't consider link travel times constant but time dependent, in other words, the link cost depends on the arrival time at the origin of the link, that is the considered shortest paths are dynamic or time-dependent (TDSP).

In this work, we have implemented a variation of the DOT (Decreasing Order of Time) TDSP algorithm proposed by (Barceló et al. 2013). The authors present an implementation of the DOT TDSP by (Chabini 1998) using the structures proposed by (Ziliaskopoulos & Mahmassani 1993). The main features of this solution are the reduction of the time achieved using the DOT concept and the reduction of in-memory space by using the Yale format for Sparse Matrices.

A detailed description of the algorithm can be found in the reference (Barceló et al. 2013)

7. Real-Time Traffic Information System

The computational framework to test the Multiple Passenger Ridesharing System described in Section 3 and its implementation depicted in the conceptual diagram of Fig. 3, assume that the fleet management approach and the core component of the Dynamic Router and Scheduler were embedded into dynamic traffic simulation model emulating the traffic conditions in a realistic way.

The logics of the process is the following:

1. Selection of a time interval step to check the feasibility of the current routes.
2. At each time interval step, the simulator sends to the link costs data base the current travel times of each link of the network.
3. With this new information, the time-dependent shortest path module re-computes the time-dependent shortest paths between all the nodes of the network at each departure time interval.
4. With these new time-dependent shortest path costs, the time dependent shortest path data base is updated.
5. The feasibility check method is launched using this new information stored in the time-dependent shortest path database. This method:
 - Recalculates the total cost of all the current routes.
 - Modifies the current solution to solve the feasibility problems.

Such dynamic traffic simulation plays a twofold role; the exchange of information required by the logics of the decision made by the fleet dispatcher and the real-time information, including link travel times, required by the dynamic routing algorithms. In other words, in a real life system we would consider that the fleet manager has access to an Advanced Traffic Information System (ATIS) supplying such information. In the computational

implementation of the system we have considered that the emulation of such ATIS was also the function of the dynamic traffic simulator. Therefore the fleet manager has access to such information and the vehicle routing algorithms are able of “continuously” checking the traffic conditions on the network as well as the fleet expected timing. In this way, the algorithm is able to manage sudden changes in real traffic travel times.

Thus to properly emulate the interdependencies and data flows between the dynamic simulation model emulating the ATIS, and the algorithms in the Dynamic router and Scheduler, some changes, and additions, must be made in the proposed algorithms.

7.1. Time interval step

Let K denote the length of the time interval of shortest path calculations, the smaller K the higher the frequency at which shortest paths are updated and therefore they accuracy in terms of traffic conditions will be higher, at the price of substantially increasing the computational costs. In consequence, efficient real-time applications must find a suitable balance between the quality of the paths and the computational burden.

It is a very important factor to take into account when generating the information as we need to balance the trade-off between precision and storage capacity.

- If precision is needed. Simulation statistics have to be computed for short time interval steps but this might require larger storage capacity (in particular for large road networks).
- If we define larger time interval steps, storage capacity will remain in reasonable levels but we could be losing important information as changes in road densities or traveltimes.

7.2. Time-dependent link cost data base

The availability of current and forecasted traffic condition from an ATIS makes possible a more efficient implementation of the shortest paths module taking into account the short and long-term forecasts of travel times, to perform the corresponding calculations. But efficient forecasts require not only the current measurements but also access to past records, in other words, to an historical time-dependent link cost data base and its corresponding update. The creation and updating of this database asks for an additional function that must be added to the computational framework:

- **Historical time-dependent link cost data base**

An off-line dynamic traffic simulation is used to generate the historical traffic database through the simulation of multiple replications.

- **Update of the time-dependent link cost data base**

Let τ be the forecast time limit of an ATIS. We will assume that this forecast is accurate enough to predict the road conditions for the following τ time units. We have called this computations as the short-term forecasts of the system.

Once the historical data base has been generated, multiple techniques can be applied to forecast the conditions of the network beyond the forecast limit γ .

When the time-dependent traffic information database (link travel times Database) is updated only a fraction of the total records is subject to changes due to the new current traffic conditions and it is represented by the new short-term travel time forecasts.

If the forecasts database is updated at time t_0 , the time-dependent travel times values that are likely to change are in the short-term interval $\{t_0, t_0 + 1, \dots, t_0 + \gamma\}$, while for $t > t_0 + \gamma$, the corresponding travel times values remain unchanged.

So, in summary, the simulation of the performance of a fleet uses two types of data involved in the computation of the shortest paths and the corresponding travel times:

- The short-term travel time forecast is the estimated travel time for the following τ time units in all links of the network when the departure time of a vehicle is less than or equal to γ .
 - Computed from the current traffic conditions supplied by the recorded statistics of the ongoing simulation replication.

- The long-term travel time forecast includes the estimated travel times when a vehicle departs at some $t > \gamma$.
 → Computed from the historical information of the network. It is stored in the historical database that has been built from multiple replications.

The above approach is illustrated on Fig. 5. Let us assume that we wish to calculate the shortest path between node i and node j . The path consists of a sequential series of links, each of them having travel times as a function of the time of arrival of the vehicle at the beginning of the link. If a vehicle is to depart from client i at time t_i , the traffic information system will provide a shortest path from client i to client j by considering the short-term forecasts for every $t \leq t_i + \tau$, which are stored in the current replication database, and the long-term forecasts for every $t > t_i + \tau$, stored in the historical database.

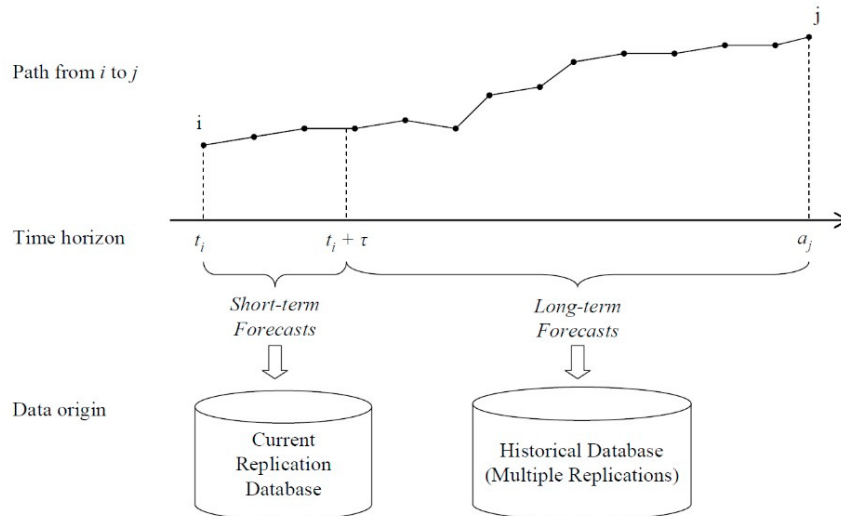


Fig. 5 Short-Term and Long-Term Forecasts in Time-Dependent shortest paths (Orozco 2010)

In this module, the key requirement from the dynamic traffic simulation software is the capability of it to generate and store traffic information in time interval steps. At each time interval step, the simulator sends to the link costs data base the current travel times of each link of the network.

7.3. Re-computing time-dependent shortest path

With the update of the time-dependent link travel costs, the time-dependent shortest path module re-computes the time-dependent shortest paths between all the nodes of the network at each departure time interval.

7.4. Feasibility check module

The objective of the feasibility check module is to manage sudden changes in traffic travel times. I.e., it checks if the routes previously assigned to the cooperative vehicles are feasible after changes in traffic conditions.

This module is launched using the updated information stored in the time-dependent shortest path data base.

- **Recalculation of the cost of the current routes**

To check the current status and attributes of each vehicle, that is its position on the road network and its occupancy. Additionally, the simulator checks which customers assigned to each of the cooperative vehicles have already been fulfilled or visited. The shortest path recalculation is then performed over the unvisited customers and the current route is updated if the route becomes unfeasible with respect to the time windows constraints.

- **Modifications to solve the unfeasibility**

If some of the current vehicle routes become unfeasible, then, this module recalculates the assigned shortest path between the customers.

8. Design of Experiments

8.1. Model Description

The selected microsimulation scenario was Barcelona's CBD (see Fig. 6), comprising 7.46 km² with more than 250,000 inhabitants. Its Aimsun (Transport Simulation Systems 2013) model consists of 1720 links, 528 nodes, 120/130 generation and attraction centroids and 877 non-zero OD pairs. The horizon study is 30 min, accounting for a total number of 20,700 vehicle trips.



Fig. 6 L'Eixample in Barcelona (left) - simulation test-site (central) and Aimsun model (right).

The microscopic traffic simulator model realistically emulates the progress of individual vehicles in the network each half second, belonging to vehicle several classes: regular cars, bus units and ride-sharing vehicles are considered. Regular car demand is modelled as 15-min time-sliced demand whose Origin-Destination pattern reproduces the actual working day morning period in Eixample district and reproduces actual dynamic traffic conditions for the modelled period. Bus units are considered along 50 bus network itineraries with frequencies and stops for boarding/alighting that are taken into account and finally, a fleet of multiple passenger ride-sharing units that are moved over the network according to the dynamic routing and scheduling to serve passenger requests. The fleet of shared units dispatching considers time-dependent routing algorithms according to real-time emulated traffic conditions provided by the simulator.

8.2. Methodological Proposal

The goal was to evaluate the new transport mode in a urban area with regard to the impacts on urban traffic and taking into account a very detailed description of the characteristics of the mobility service. The implementation of mobility strategies into a specific real scenario was considered. The system was design to be flexible to host a wide range of service possibility in the sense that some quantitative parameters can be set:

- as fleet size, vehicle capacity in the operational features set,
- the market quota for the new mode as a percentage of the total demand for the study period,

And also, some qualitative characteristics of the new mode depending on the mobility strategy and seriously affecting the algorithmic implementation and the design of the underlying VRP, as

- Fully flexible routes, fixed routes or mixed routing
- Boarding/Alighting allowed either everywhere in the network or at pre-defined stops or regular-bus stops or any chamfer of Eixample geometry
- Depot existence: number and location
- Pre-booked demand service available or not

- Combined service for passenger and freight distribution
- Use of reserved bus lanes

Currently, all the indicated service and operational strategies are not implemented, but they will be in the future. The involved factors in the preliminary design of experiments that have been considered are:

- **Global Traffic Demand Scenarios:** Several options were initially considered such as Rush hour (morning/afternoon peak in working day), No Rush hour, Special Event (to be specified) and Incident (to be specified). Intensive testing has been performed in the rush-hour scenario since special event and incident scenarios are too dependent of external attributes for being useful in this stage of assessment of the new mode.
- **Demand Request Generation:** For each of the three defined types of customer requests (pre-booked requests for the beginning of the considered journey, pre-booked requests for other time of the considered journey and on-time demand). The demand is modeled to be generated in two different ways: requests following the same pattern than the demand of private vehicles and requests randomly distributed over the site geometry. Testing has been performed using spatially generated demand following the same pattern than the demand of private vehicles in such a way that for each origin and destination centroids, the corresponding percentage of the global demand (depending on the market quota of the new mode design factor, 5, 10 or 15%) is generated over a circle with a ratio of 200 meters around the centroid. In order to avoid short trips the minimum travel distance can be set in the settings of the system and a minimum length of 0.5 km was set. No demand behavioral analysis and modal split has been undertaken, hence the total number of passengers assumed to be served by the mode has been selected between 5%, 10% or 15% of the total auto demand in the study period. The passenger's initial and final position and time window flexibility either at initial/final or both points of the route have been considered leading to model an ultra-flexible service strategy. Fig. 7 shows an example of the distribution over the site of 1000 requests (origins in red and destinations in green).

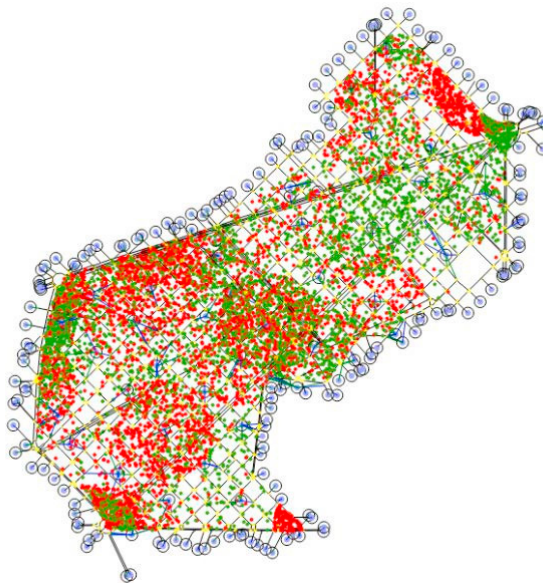


Fig. 7. Spatial distribution of the generated demand for the new mode.

- **Fleet dimension:** it was considered between 250, 500, 750 and 1000 vehicles. A no depot solution was considered and fleet vehicles are parked in specific allowed zones at the beginning/end of the passengers' transport journey, these emulated zones are the regular loading/unloading area for freight distribution in the chamfers across the road network of Example. When a new mode's unit is not operating (i.e. it is empty and it is

waiting for the arrival of new request), it goes and parks in the allowed zones until a new requested demand has to be served. Fleet is allowed to use bus restricted lanes and also to pick up and deliver passenger in bus-stops.

- **Capacity of the units** that operate in the new mode. No mixed characteristics were taken into account, but seated capacities from 6 to 10 passengers were assessed. A length of mean 5 m with min and maximum unit length of 4.6 and 5.5 m normally distributed is considered. A maximum desired speed between 80 and 110/km/h is considered, but a smoother acceleration/deceleration parameters for car-following models is set. Urban speed limitations are taken into account in car-following models as in regular vehicles.
- **Time Window Width:** The width of the time interval in which the demand requests must be served (pickup and delivery time windows). Considered levels are 10, 15 and 30 min and the base for comparison where no restriction is assumed.
- **Degree of Dynamism:** The % with respect to the total demand for the study period that is not pre-booked in advance (% real time requests). Considered levels have been 0%, 20%, 50%, 80% and 100%. The “Customer Request Emulator” is an API specifically developed to generate three types of new mode demand requests (on-time demand, pre-booked demand to be served when the simulation starts, and pre-booked demand to be served at a certain specific time). The percentage of each of these demands was defined to be modified in the settings of the system.

Aimsun’s result tables account for statistical performance measures for each user-defined interval of time splitting results into vehicle classes. Nevertheless, the new mode has a fleet of vehicles whose routes do not have an origin-destination, but correspond to a vehicle routing problem (using dynamic traffic conditions) optimizing the overall requests using the available fleet size and capacity, real-time traffic conditions and time-window limitations. Thus an *ad hoc* module was designed and implemented to gather all the data from the new mode operations required to calculate the performance measures. A specific Redis open-source database collected for every scenario and replication data allowing to calculate from the passenger, the mode operator and the local authorities the performance indicators. For each simulation replica, fleet unit and simulation step, the position of each unit, the number of passenger is calculated.

From the passenger point of view the performance indicators that have been defined are:

- Average travel time per passenger (it includes in-vehicle plus waiting time).
- Average passenger’s speed
- Average in-vehicle time per passenger.
- Average waiting time.
- Distance travel by fleet units (km/h)
- Rejected requests

From the point of view of the operator:

- Number of person-trips per hour
- Average number of persons per vehicle-unit
- Average number of occupied vehicles
- Total vehicle-km
- Total vehicle-hour travel times

From the local authorities’ point of view:

- Use of traffic area
- Traffic congestion measures at system level: average density, flow and speed.
- Potential for connection with other transportation systems

For each replication, statistical data was stored in the default Aimsun database and the specific Redis DB designed to collect data for multiple ride-sharing service.

8.3. Analysis of Simulation results

The computational burden depends roughly on the fleet size and the total number of requests, leading to computing times for each replication around 10 hours. To reduce the computational burden a fixed number of 5 replications for each scenario was considered after testing in the base scenario the requested relative precision of less than 10% in the average passenger travel time performance indicator.

A graphic representation of a fixed replica can be obtained postprocessing the adhoc Redis database for fleet collecting data during the simulation is shown in Fig. 8: fleet vehicles have been represented in different color depending on the number of passengers inside the vehicle as in the example displayed in. where red dots indicate fleet units with 0 passengers, brown dots 1 to 2 passengers, orange 3 to 5 passengers and green more than 6 passengers.

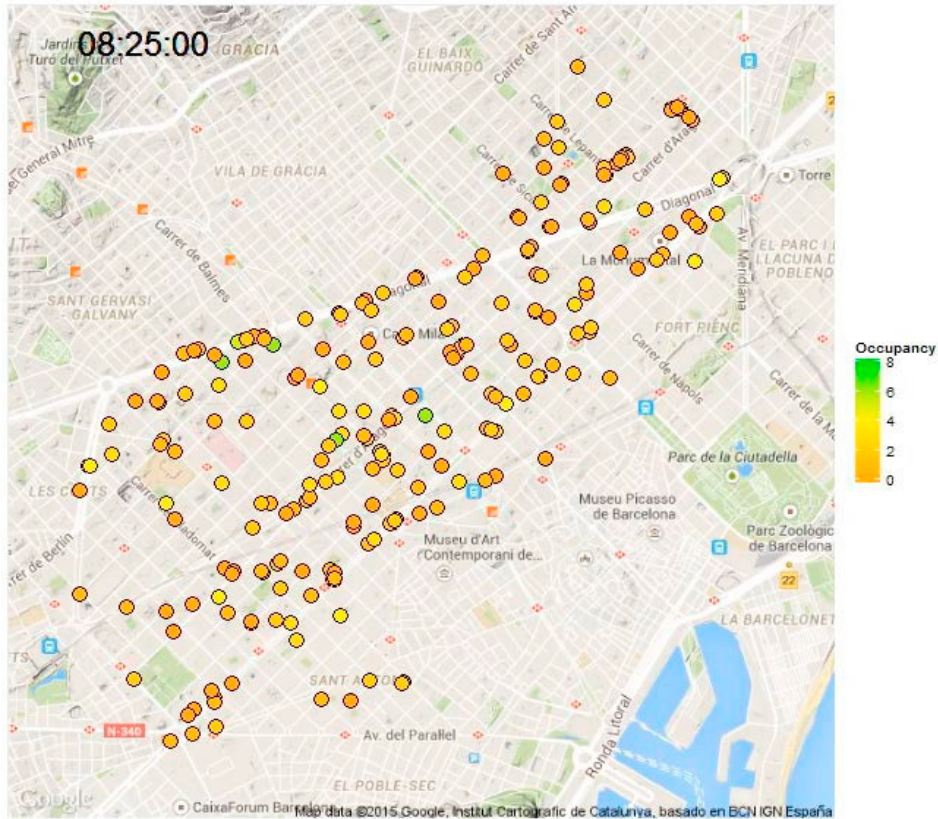


Fig. 8. Car occupancy graphical output for a simulation replica and step on Google Map

Remarkable results account for average travel time per passenger and average waiting time vs fleet dimension from the passenger point of view. The overage occupancy of fleet units and the percentage of accepted demand according to fleet size and time-window constraints are critical performance indicators from the operational point of view. Fig. 9 supports **a fleet size of 500 vehicles can serve a demand of up to 5% rush hour demand** of Barcelona Example with a ratio of accepted requests estimated of more than 90% and accounting for a total number 2600-2800 passengers, while the acceptance rate reduces about 80% for 5000-5200 passenger requests representing 10% of rush hour demand. Pre-booked demand was set to 50% of the market quota for the new mode and the generation was spread over the first 30 min of the simulation horizon (base case).

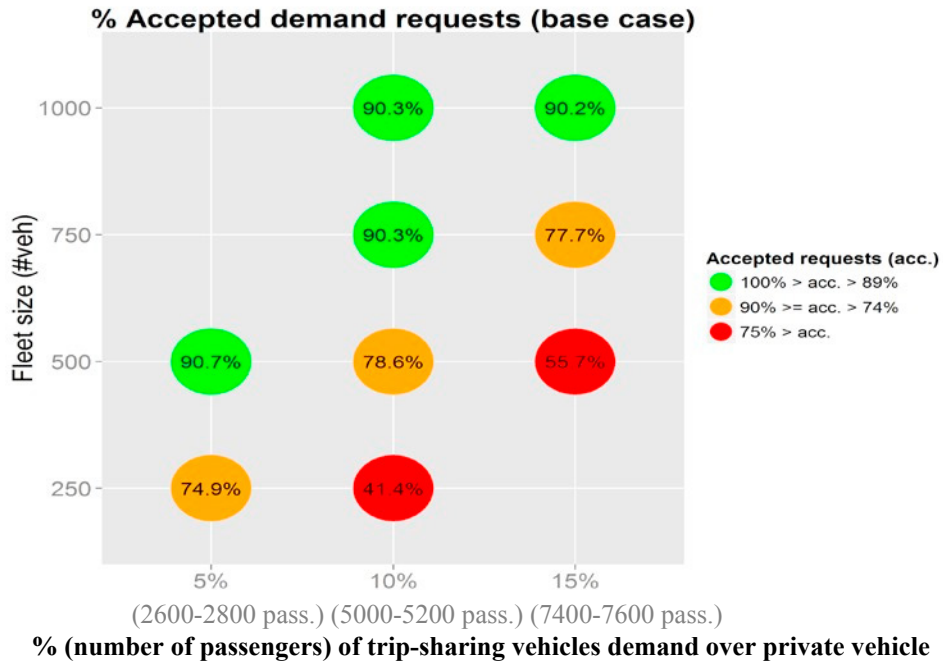


Fig. 9. Demand acceptance subject to time-window restrictions

In Fig. 10 on the right, the number of passengers that are travelling in the new mode at each instant of the simulation time (one hour) is shown in the base case scenario that assumes a fleet size of 500 units, time windows of 15 min and a prebooked demand of 50%. There are different lines for each of the percentage of auto demand assumed for the new mode: 5%, 10% and 15%. In this case, we can see that after the start-up period (approximately 20 min), the new mode becomes stable with respect to the number of passengers using it (approximately 750 passengers at each instant of time).

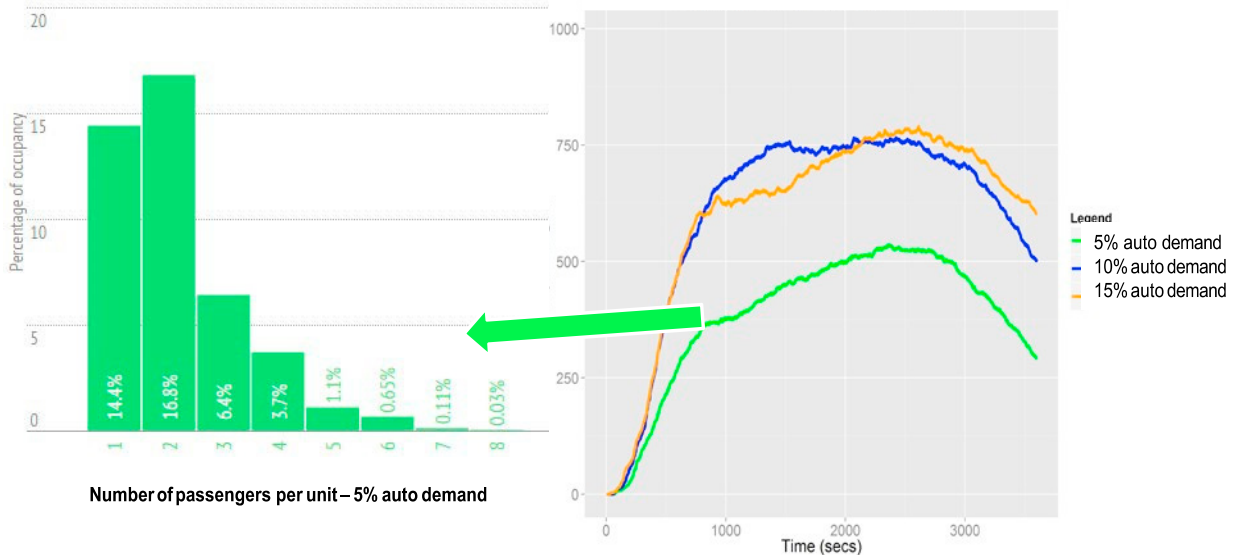


Fig. 10. Total Passengers over time (right) for base-case. Distribution of shared vehicle occupancy for the 5% demand

On the left of Fig. 10, the distribution of the number of passengers per unit, occupancy, is depicted for the 5% market quota of the new mode. We differentiate among the units with 1 or more passengers into, the units that are empty but are in service since they are going to pickup someone. In the particular case, of 5% of market quota, approximately 43% of the units are occupied. The average occupancy of non-empty units is 2.14 passengers, but including units that are empty, but going to pickup someone then the occupancy average in this case is 1.54 passengers. In any case, the occupancy is quite low due to the particular simulation model considered where the weighted trip length is less than 2km. At simulation time 0, vehicle units for service are assumed to be in their assigned parking lots, so pick-up for the first customer of the day requires some time (start-up period).

9. Conclusions and Future Research

The approach has demonstrated the potential feasibility of the some proposed mobility concept for on demand multiple passenger ride-sharing in urban area. The main outcomes are:

- An architecture has been built on top Aimsun microscopic traffic simulator to be able to emulate and evaluate general policies of this new mobility service.
- Result analysis for some preliminary experimental settings show, among other results, that the new transport mode is beneficial for cities in terms of reducing traffic flow and emissions and also beneficial for users in terms of total travel time.
- Further mobility strategies for multiple passenger ride-sharing mobility concepts could be evaluated either improving current algorithms or developing variants to cope with new mode capabilities.
- A Barcelona Extended model to deal with Metropolitan area mobility has to be updated to test by simulation the many possibilities for the multiple passenger ride-sharing concept that have been developed so far.

Acknowledgments

This research was funded by TRA2014-52530-C3-3-P of the Spanish R+D National Programs and by the Secretaria d'Universitats i Recerca de la Generalitat de Catalunya under 2014 SGR 1534. Throughout, the authors have benefited from the support of Rosa María Martín, German Navarro and Antonio Rodríguez from inLab FIB.

References

- Anon, 2015. KUTSUPPLUS. Available at: <https://kutsuplus.fi>.
- Barceló, J., Orozco, J.-A. & Grzybowska, H., 2013. Making Real-Time Fleet Management Decisions Under Time-Dependent Conditions in Urban Freight Distribution. In E. V. de V. (ed. . Moshe Ben-Akiva , Hilde Meersman, ed. *Freight Transport Modelling*. pp. 453–484.
- Boesch, M.P., Ciari, F. & Axhausen, K.W., 2016. Required autonomous vehicle fleet sizes to serve different levels of demand. In *Proceedings of the 94th Transportation Research Board Annual Meeting*. Washington D.C., p. Paper 16–2682.
- Campbell, A.M. & Savelsbergh, M., 2004. Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. *Transportation Science*, 38(3), pp.369–378.
- Chabini, I., 1998. Discrete Dynamic Shortest Path Problems in Transportation Applications: Complexity and Algorithms with Optimal Run Time. *Transportation Research Record: Journal of the Transportation Research Board*, 1645, pp.170–175. Available at: <http://trrjournalonline.trb.org/doi/abs/10.3141/1645-21>.
- Christofides, N., Mingozzi, A. & Toth, P., 1979. The Vehicle Routing Problem. In P. T. and C. S. (eds. . N. Christofides, R. Mingozzi, ed. *Combinatorial Optimizations*. John Wiley & Sons, New York.
- Fagnant, D. & Kockelman, K., 2014. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. *Transportation Research Part C: Emerging Technologies*, 40, p.pp 1–13. Available at: <http://dx.doi.org/10.1016/j.trc.2013.12.001> \nhttp://www.sciencedirect.com/science/article/pii/S0968090X13002581\nhttp://trid.trb.org/view/1304706.
- Furuhata, M. et al., 2013. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57, pp.28–46.
- Grzybowska, H. & Barceló, J., 2012. Decision Support System for Real-Time Urban Freight Management. *Procedia - Social and Behavioral*

- Sciences*, 39, pp.712–725. Available at: <http://www.sciencedirect.com/science/article/pii/S187704281200609X>.
- Ma, S., Zheng, Y. & Wolfson, O., 2015. Real-time city-scale taxi ridesharing. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 27(7), pp.1782–1795. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6847170.
- Martinez, L.M., Correia, G.H.A. & Viegas, J.M., 2015. An agent-based simulation model to assess the impacts of introducing a shared-taxi system: An application to Lisbon (Portugal). *Journal of Advanced Transportation*, 49(3), pp.475–495.
- OECD - International Transport Forum, 2015. Urban Mobility System Upgrade: How shared self-driving cars could change city traffic. *Corporate Partnership Board Report*, pp.1–36.
- Orozco, J.-A., 2010. *A Microscopic Traffic Simulation Based Decision Support System for Real-time Fleet Management*. Universitat Politècnica de Catalunya UPC.
- Orozco, J.-A. & Barcelo, J., 2012. Reactive and proactive routing strategies with real-time traffic information. *Procedia Social and Behavioral Sciences*, 39, pp.633–648.
- Solomon, M.M., 1987. Algorithms for the Vehicle Routing and Scheduling Problems With Time Window Constraints. *Operations Research*, 35(2), pp.254–265. Available at: <http://proxy.lib.uiowa.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=15255268>.
- Sullivant & Frost, 2015. Intelligent Mobility 3.0 Future of Mobility & New Mobility Business Models. In Sullivan and Frost, ed. London. Available at: <http://www.frost.com/sublib/display-report.do?id=MB85-01-00-00-00>.
- Transport Simulation Systems, 2013. Aimsun 7 Dynamic Simulator User's Manual.
- Ziliaskopoulos, A.K. & Mahmassani, H., 1993. Time-dependent shortest path algorithm for real-time intelligent vehicle highway system applications. *Transportation Research Record*, 1408, pp.94–100.