A Distributed Optimized Approach based on the Multi Agent Concept for the Implementation of a Real Time Carpooling Service with an Optimization Aspect on Siblings

Manel Sghaier *manel.sghaier@ec-lille.fr*
LAGIS, EC-Lille
Ecole Centrale de Lille, BP 48 - Cité Scientifique
Villeneuve d'Ascq, 59650, France

Hayfa Zgaya *hayfa.zgaya@univ-lille2.fr*
ILIS
42, Rue Ambroise Paré
59120 - LOOS, France

Slim Hammadi *slim.hammadi@ec-lille.fr*
LAGIS, EC-Lille
Ecole Centrale de Lille, BP 48 - Cité Scientifique
Villeneuve d'Ascq, 59650, France

Christian Tahon *tahon@univ-valenciennes.fr*
UVHC - LAMIH CNRS
Université de Valenciennes et du Hainaut-Cambrésis
Valenciennes Cedex, France

Abstract

Thanks to the important and increasing growth of the carpooling phenomenon throughout the world, many researchers have particularly focused their efforts on this concept. Most of the existent systems present multiple drawbacks regarding automation, functionalities, accessibility, etc. Besides, only few researchers focused on real time carpooling concept without producing promising results. To address these gaps, we introduce a novel approach called DOMARTiC: a Distributed Optimized approach based on the Multi-Agent concept for the implementation of a Real Time Carpooling service. We particularly focus on the distributed and dynamic aspect not only within the geographical network's representation but also regarding the used automatic tools and the implementing algorithms. Adequate modeling on the base of which a distributed architecture is set up has been adopted helping to perform decentralized parallel process. This helped to take into consideration different aspects we should be involved in, especially the optimization issue as users' requests must be performed in a reasonable runtime. Responses provided to users should also be efficient with regards to the fixed optimization criteria.

Keywords: Real Time Carpooling, Optimization, Network's Decomposition, Distributed Dynamic Graph Modeling, Multi-Agent System, Distributed Algorithm.

1. INTRODUCTION

With the emergence of organisms and events focusing on the emergency of addressing current environmental problems, researches have been developed to provide better conditions for survival. In particular, financial problems in addition to environmental ones caused by transport means evolution, made researchers consider this issue more in depth.
According to a study achieved in year 2000 [1], the total number of cars exceeded 740 million in the world. This car's invasion came to extend individual and collective problems (i.e. financial limitations, CO_2 emission ...) [2] importantly influencing people behavior [3]. Although public transport means remedied to almost all of these problems, they unfortunately could not afford as

much moving liberty, flexibility and comfort. Thus, tackling those problems has become increasingly urgent. Efforts developed in this context to reach a compromise, led to innovative transport services. Among them, carpooling is a concept that brought a great interest since more than a couple of decades. Indeed, several researchers dealt with carpool problem and many systems are already operational in Europe and throughout the world [2]. Technologies used in such systems and results carried out up to now are rather promising and very rewarding [4]. The fact remains that existing works show some limits. This made us consider this issue and propose a novel approach called DOMARTiC with an optimization dilemma dealing with real time users' requests and trying to perform optimized responses. Thus, in this paper, we focus on the carpooling concept, basically, with an innovative view that tackles the problem of handling instantaneously received users' requests. Our first target is then to improve the quality of service so that subscribers could obtain real time generated responses, efficiently and promptly. For this purpose, we consider a new network's representation, a conceptual modeling and the automatic tools needed to implement necessary algorithms to generate optimized carpooling service. We mainly focus on the multi-agent concept since it provides, in combination with high technologies, the efficiency required to deal with real time users' queries.

The rest of this paper is then organized as follows: in section 2, a general background provides a non-exhaustive list of the existent carpooling systems. Section 3 describes the proposed approach DOMARTiC with a focus on the multi-agent concept and the distributed assignment algorithm performed to generate optimized matching. Details are also given about the formal specifications of our problem, the optimization aspect within it and the network's decomposing process is briefly described. Section 4 comes to sum up our work and present future prospects.

2. SCRUTINY OF THE RELATED WORKS

Also known as ride-sharing or lift-sharing, carpooling refers to the shared use of a car by the driver and one or more passengers, usually for commuting[1]. Carpoolers share journeys if their personal choices match[2] (i.e. Trip origin, destination, date and time or time slots, etc.), (Figure 1).

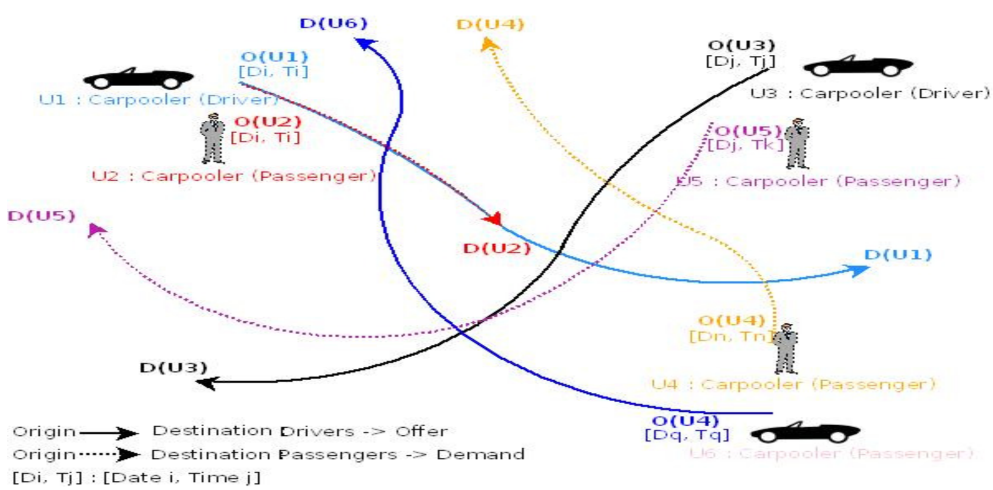


FIGURE 1: Carpooling : Offers and Demands.

Largely due to its multiple advantages [4], carpooling has become in recent years a remarkable phenomenon and individuals increasingly tend to appreciate it. In this context, urban carpooling is often promoted as an alternative to owning a car [5], [6]. Besides, carpooling combats rising traffic congestion [4], reduces energy consumption and moving costs.

[1] http://www.answers.com/topic/carpool
[2] http://www.carpoolglobal.com

Regarding its promising impact on people and environment, it was considered in many works [2] and formal carpool projects have been around in a structured form since the mid-1970s. They are rather shown as:

- Virtual supports for reservations' management such as 123envoiture[3] [4] or aide-covoiturage.com[5] that allow users to subscribe in order to access carpooling services and post specifications about their trips or consult planned trips and join each other if one of the available proposals meet their needs.
- Virtual social networks or forums (e.g. COMOVE[6]) that propose automatic tools allowing their subscribers to directly meet and agree on details about trips to share. Forums suit for the same principle as the one detailed above since they are presented as websites where, besides of consulting drivers' offers and passengers' requests, people can meet and directly discuss through chat-rooms.

Despite the important progress experimented on carpooling concept thanks to the existent works, it still remains in embryonic stage regarding automation and real time aspects. Indeed, existing systems only consider the standard deal with offers and requests storage. Almost all of these works only tackle the static ridesharing issue whereby users must plan their trips in advance (Figure 2) and so neglecting the dynamic aspect.

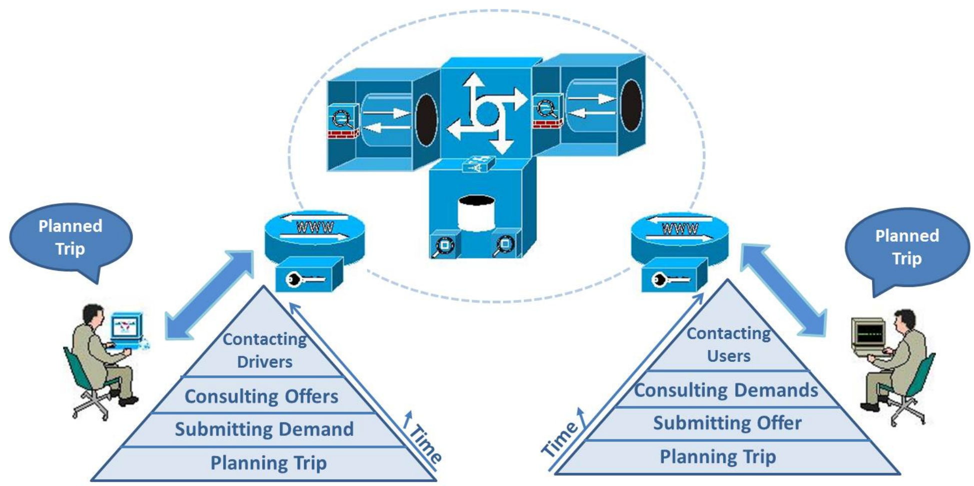


FIGURE 2: Carpooling: Existing Systems.

In order to make up this shortfall, we propose to consider the dynamic aspect. In fact, even if real time carpooling was considered in some approaches (e.g. GoLoco, Easy-Rider, T.écovoiturage…) [2, 7], most of them no longer exist. Furthermore, none of the existent systems evoked the security issue which is one of the main reasons hindering their success. Thus, we are especially involved in the concepts of traceability, communication and security services.

3. DOMARTIC: AN OPTIMIZED PROCESS TO PROVIDE A REAL TIME CARPOOLING SERVICE

Setting up a ridesharing service as satisfactory as possible is our main objective. Proceeding from this basic principle and in an attempt to remedy the problems outlined above, we consider the problem of processing real time users' requests. The proposed approach mainly aims at carrying out a system that allows its users to reach a vehicle anywhere, at any time and as rapidly

[3] http://www.greencove.fr
[4] http://www.123envoiture.com/
[5] http://www.aide-covoiturage.com
[6] http://www.comove.com/forum-covoiturage/index.php

as possible. For this purpose, adequate tools as well as an appropriate solving methodology are considered. Among them, a based GPS positioning tool as we need to be provided with real time information about drivers and passengers so that we can process their requests and perform vehicles' traceability and so ensure users' safety. To do this, system users must have already agreed on a number of terms of use. For example, they must accept to be located at any time of the trip. This is done first time they access the system so that they can subscribe.

As the technological development is at its peak (e.g. GPS geolocation, mobile devices connected via GPRS or Internet…), it is convenient to take advantage from the observed advancement especially in networks and communication to set up a highly interactive framework. The elaborated automated support enables users to communicate with each other and with the system itself. Figure 3 illustrates a real view of our system showing possible communications between its actors. In addition to users, vehicles and the GPS module, the assignment process constitutes a main and central actor and has a pivotal role ensuring communication with and between the other actors. As it is responsible for automatically processing dynamic and optimized allocation of vehicles to users, it is considered as the core of our work.

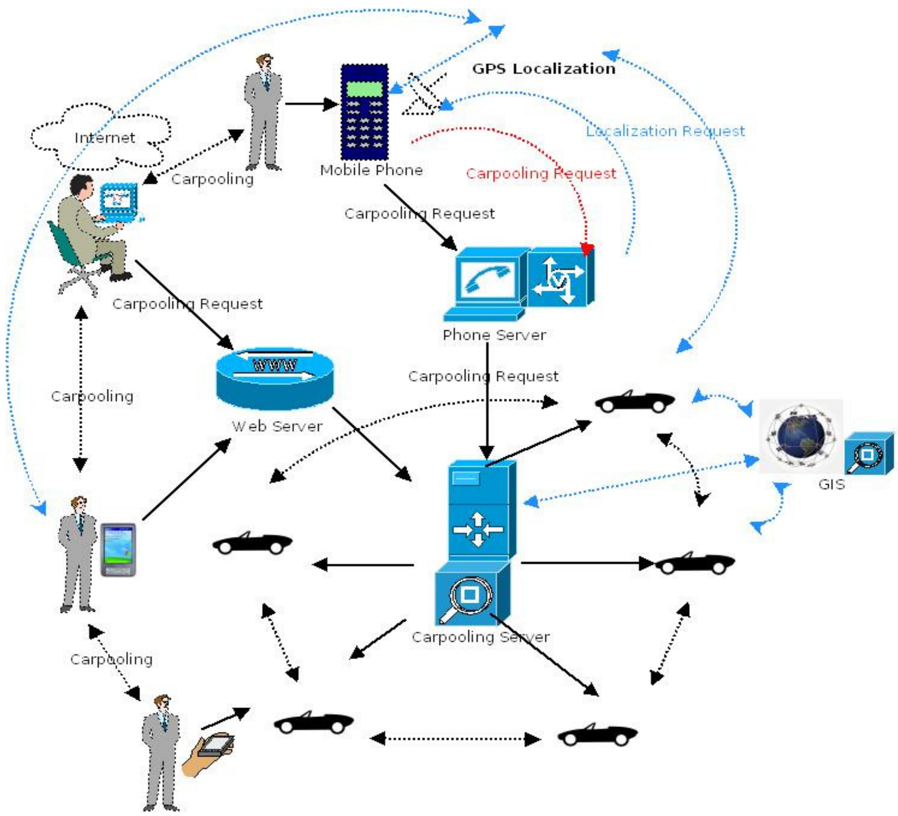


FIGURE 3: A based Communication Support within DOMARTiC.

In this context, our work is directly involved in setting up a complete carpooling framework that primarily establishes several functionalities (see Figure 4). Among the latter, parallel requests acquisition and their parallel decentralized process according to a distributed architecture set up through a subdivision process lately defined.
Optimization is the key word in the process developed and is considered on more than one sight:
Firstly considering the optimized requests' process according to the chosen criteria,
And secondly considering the real time constraints according to which responses must be provided within an acceptable time of processing (i.e. as minimum as possible).
Consequently, we are mainly concerned with providing, within a reasonable response delay, the best solution or at least an optimized or an approximate one.

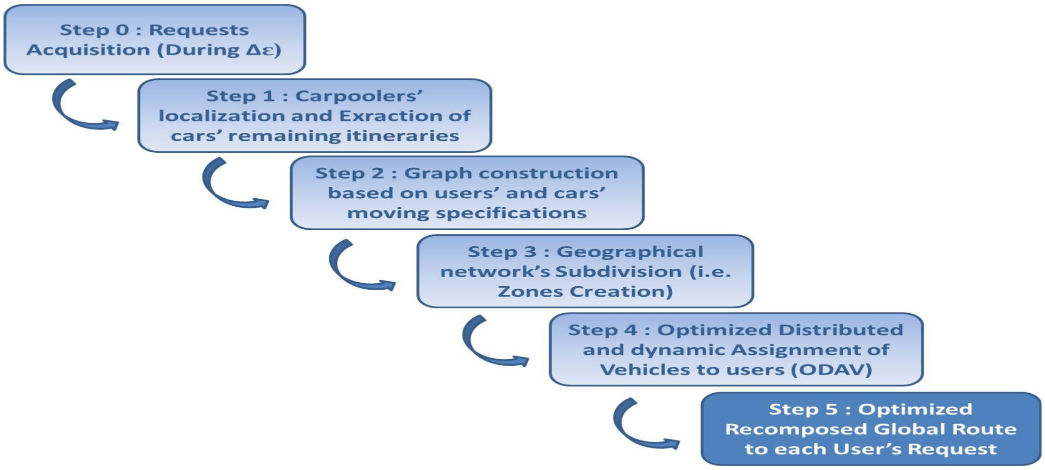


FIGURE 4: DOMARTiC's Main Functionalities to establish an Optimized Process.

3.1. Parallel Requests' Processing

Setting up a dynamic ridesharing system should instantly take into account users' requests for immediate trips. Thus, to guaranty users' satisfaction regarding required optimum response delay, we are involved in instantaneously process their requests. Moreover, users may probably issue queries approximately at the same time which should be processed in parallel. For this purpose, we introduce a parameter called $\Delta\varepsilon$ [8] that indicates a negligible time lapse during which requests' acquisition should be performed (Figure 5).

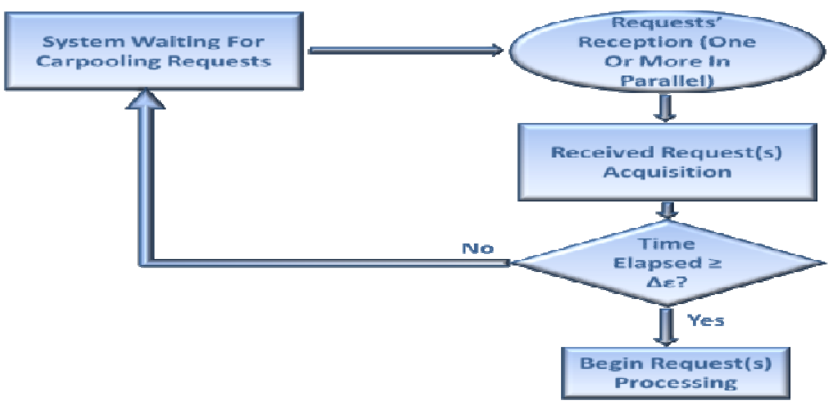


FIGURE 5: Parallel Requests Processing within an Optimized Real Time Carpooling System.

3.2. Optimization Within DOMARTiC: Towards a better Quality of Service

Considering the dynamic carpooling issue involves optimization with a driving idea that refers to users' satisfaction. As shown in figure 6, multiple criteria could be considered for this purpose.

FIGURE 6: Optimization Criteria to consider in Dynamic Carpooling Services.

Optimization mainly concerns vehicles' dynamic allocation to passengers, searching for the best offer meeting users requests' specifications, or at least an approximate proposal.
In this paper, optimization primarily lies in selecting solutions maximizing comfort (i.e. minimizing the number of transfers) trying by the way to ensure trips' continuity. In this context, vehicles serving a given route asked by a given user are considered if and only if they satisfy a set of feasibility constraints (lately formulated). These constraints mainly refer to the number of places available, times of departure and arrival, itineraries of the considered cars that must fit the asked route while matching the user's moving preferences (i.e. departure and arrival time, number of persons…) on this route.

Secondly, an optimal solution is chosen among the firstly selected ones considering another criterion. Indeed, the final selection focuses on the solution that minimizes the global Trip Duration. The latter involves both of the Waiting time (W_t) and the Traveling time (T_t). Thus, a Fitness function aggregating W_t and T_t is considered in a multiobjective optimization optical view.

Complexity of the Optimized Real Time Carpooling Problem (ORTCP)
Considering an optimized dynamic carpooling system implies to consider the problem of its combinatorial complexity. To prove ORTCP's high complexity, we consider a comparative study (Table 1) with the *Dynamic Pick-up and Delivery Problems* (*DPDPs*), especially the *Swapping Problem* (*SP*) [9] or *Dial-a-Ride problem* (*DARP*). *PDPs* are a class of vehicle routing problems in which objects or people have to be transported between an origin and a destination [10].

Problem	Parameters	Constraints	Objective Function
DPDPs: Dynamic Pick-up and Delivery Problems	Vertices with objects or persons (Dynamic Swapping Problem *SP* or Dial-a-Ride Problem *DARP*)	Vehicle with limited unit capacity, real time requests…	Search for the shortest path to accomplish the rearrangement of the objects (or persons)
Optimized Real Time Carpooling Problem (*ORTCP*)	Users' Locations, Requests' Specifications (Origins and Destinations), Intersections with Cars' Routes	Cars with limited number of available places, cars' itineraries, real time requests…	Search For an Optimized itinerary regarding Trip Duration For Each User's Request

TABLE 1: Dynamic Carpooling Problem in comparison with *SP and DARP*.

ORTCP is obviously analogous to *DPDPs*. The *SP* as well as *DARP* are known to be of exponential complexity and belong to the NP-Hard problems category. So, this comes to confirm the high complexity of *ORTCP* and states it as a combinatorial optimization problem (i.e. O(*ORTCP*) ≡ O(*SP*) ≡ O(*DARP*)). Moreover, many other aspects have not been considered in the previous study and are liable to heighten even more the complexity. Among those aspects:

- ORTCP should also take into account cars' moving specifications,
- There is more than one car that could respond to one or more request …

Jointly solving those problems become even more complex as much as there is more users' requests and/or more available cars (i.e. O (*ORTCP*) > O (*SP*)).
This study reveals many difficulties that would probably present a big handicap through the way to process optimization tasks. Thus, we propose to decompose the original problem into partial ones to promote distributed parallel process.

3.3. A Subdivision Principle to set up a Distributed Dynamic Architecture

In order to overcome *ORTCP*'s high complexity, we propose a novel network representation mainly concerned with the way to elaborate a distributed architecture trying to divide the initial whole task into smaller parallel ones and so reduce the initial problem's complexity. Proceeding from this basic concept, we might be able to perform distributed parallel requests processing.

DOMARTiC's Formal Specification

Partially inspired by the PDPs and the Vehicle Routing Problem (VRP) formal modeling [11], we define the dynamic optimized carpooling problem, at a given time *t*, as the triplet $T(t) = (G(t), D(t), O(t))$ where:

- $G(t) = (N(t), A(t))$ is a directed graph with :
  - $N(t)$: The set of nodes set up on the base of geographical coordinates corresponding to origins, destinations or Intermediate Destinations of either asked or offered trips at time *t*.
  - $A(t)$: *A* set of directed arcs that represent real time constructed paths relating origins to their destinations. They concern vehicles' remaining itineraries and users' asked routes in function of existent intermediate destinations.

  $G(t)$ is real time constructed and has no fixed characteristics or shape since it mainly depends on users' requests (i.e. $D(t)$) and cars' offers (i.e. $O(t)$). Consequently, $G(t)$ is a dynamic graph.

- $D(t) = \bigcup_{p=1}^{n} D_{U_p}(t)$ is the set of carpooling demands received at time *t*. It refers to $U(t) = \bigcup_{p=1}^{n} (U_p(t))$: the set of n users having issued requests. Each user U_p's request is defined as $D_{U_p}(t) = (d^+_{U_p}, d^-_{U_p}, P_{U_p}, [De_{U_p}, A_{U_p}])$ where :
  - $d^+_{U_p}$ is the origin of the required route. It may refer to his geographical localization whenever it is not specified by the user U_p.
  - $d^-_{U_p}$ is the destination of this route. Moving origins and destinations are represented by a couple of nodes in the graph $G(t)$. Thus, $N_D(t)$ is the set of nodes in G specific to users' demands:

  $$N_D(t) = \bigcup_{p=1}^{n} (d^+_{U_p}, d^-_{U_p})\ (t) \subset N(t)$$

  - P_{U_p} is the number of Persons including U_p, that must move from $d^+_{U_p}$ to $d^-_{U_p}$.
  - $[De_{U_p}, A_{U_p}]$ designates a limited time interval that defines the global trip duration.

De_{U_p} is the Departure time at the earliest, and A_{U_p} is the time of Arrival at the latest. This interval, when specified, represents the preferred departure time and at worst the tolerated Arrival time for a given request. Otherwise, the concerned user may have no preferences and do not specify one of these parameters or both. In this case, default values of De_{U_p} and A_{U_p} are respectively the current time and an Estimated Arrival Time (EAT) calculated as follows:

$$EAT(d^+_{U_p}, d^-_{U_p}, t) = TT(d^+_{U_p}, d^-_{U_p}, t) + TDT(d^+_{U_p}, d^-_{U_p})$$

Where the Traveling Time $(TT(O, D, t))$ is the function weighting arcs of G and calculating the time needed to traverse a path from a given origin O to a determined destination D:

$$TT(O, D, t) = Distance(O, D) * TAT(O, D, t) / (O, D) \in A(t)$$

and $TDT(O, D)$ is the Tolerable Delay Threshold apportioned to a fixed tolerable delay of 10 minutes per 50 kilometers. It represents the acceptable maximum delay on the distance between O and D:

$$TDT(O, D) = Distance(O, D) * \frac{10}{50}$$

Distance (O, D) is the formula calculating distance between two GPS coordinates:

$$Distance(O, D) = \sqrt{(Longitude_O - Longitude_D)^2 + (Latitude_O - Latitude_D)^2}$$

TAT is a given data on the Traveling Average Time that depends on roads' type, the period of traveling (e.g. rush hour, normal period...) and the weather. Then, arcs' weights could vary over time. Based on this and on its real time defined specifications (i.e. arcs and nodes), we can state that G(t) is a highly dynamic graph. A graph is dynamic if at least one of its parameters is function of time [12][13].

As two users or more could ask for a same route, the set of requests D (t) is then modeled as an Origin to Destination $OD_U(t)$ matrix (k x k):

$$D(t) = \begin{array}{c c} & \begin{array}{ccccc} d_1^- & \cdots & d_i^- & \cdots & d_k^- \end{array} \\ \begin{array}{c} d_1^+ \\ \vdots \\ d_i^+ \\ \vdots \\ d_k^+ \end{array} & \begin{pmatrix} D_{(d_1^+, d_1^-)} & & \cdots & & \cdots \\ \cdots & & \cdots & & \\ \vdots & & D_{(d_i^+, d_i^-)} & & \vdots \\ & & & & \\ \cdots & & \cdots & & D_{(d_k^+, d_k^-)} \end{pmatrix} \end{array}$$

Each element $D_{(d_i^+, d_i^-)} / i \in \{1..k\}$ of the matrix refers to the set of users' demands on a given route i defined by its origin d_i^+ and destination d_i^-. As there could be more than one user asking for the same route (d_i^+, d_i^-), matrix's element referring to the latter contains the set of users' demands on this route showing the different parameters (i.e. number of persons, time of departure at the earliest, and the arrival time required) specific to each one of them.

$O(t) = \bigcup_{j=1}^{m} O_{V_j}(t)$ is the set of cars' offers. $V(t) = \bigcup_{j=1}^{m} (V_j(t))$ is a fleet of m vehicles offering journeys and already circulating through the geographical network at time *t* or not yet. Each vehicle V_j has a limited capacity $C_{V_j} \in C$ and its origin $(O^+_{V_j})$ and destination $(O^-_{V_j})$ are designated by a couple of nodes in $G(t)$:

$$N_O(t) = \bigcup_{j=1}^{m} (O^+_{V_j}, O^-_{V_j})(t) \subset N(t)$$

$N_O(t)$ is the set of offered trips' specifications related to cars' origins and destinations.

$O_{V_j}(t) = (O^+_{V_j}, O^-_{V_j}, PL_{V_j}, \begin{pmatrix} ID_1 \\ \cdots \\ ID_z \end{pmatrix}, [De_{V_j}, A_{V_j}])$ is V_j's offer where besides of parameters $(O^+_{V_j}, O^-_{V_j}, [De_{V_j}, A_{V_j}])$ similar to those defined in requests, the driver may specify:

- Pl_{V_j} : the number of places available in V_j at time *t* for a given route (i.e. $[O^+_{V_j}, O^-_{V_j}]$).
- $ID_{V_j} = \begin{pmatrix} ID_1 \\ \vdots \\ ID_z \end{pmatrix}_{V_j} = \bigcup_{q=1}^{z} ID_{q,V_j}$: The set of addresses V_j would pass by to reach its final destination. These addresses are defined as Intermediate Destinations (*ID*) that can be either specified by the driver or set by the system in previous process as pickup or deposit places and represent a set of specific vertices $N_{ID}(t)$ of $G(t)$:

$$N_{ID}(t) = \bigcup_{j=1}^{m} \bigcup_{q=1}^{z} ID_{q,V_j}(t) \subset N(t)$$

Thus,

$$N(t) = N_D(t) \cup N_O(t) \bigcup N_{ID}(t)$$

The set of offers (i.e. O(t)) is modeled as an Origin to Destination (i.e. OD_V (t)) matrix that considers each single route $[o,d]$ / $(o,d) \in (\bigcup_{i=1}^{f} O_i = N_O(t) \cup N_{ID}(t))^2$.

$$O(t) = \begin{array}{c|c} & \begin{matrix} O_1^- & \ldots & O_i^- \ldots & O_f^- \end{matrix} \\ \begin{matrix} O_1^+ \\ \vdots \\ O_i^+ \\ \vdots \\ O_f^+ \end{matrix} & \begin{pmatrix} O_{(O_1^+, O_1^-)} & \cdots & \cdots \\ \cdots & \cdots & * \\ \vdots & O_{(O_i^+, O_i^-)} & \vdots \\ * & & X \\ X & \cdots & O_{(O_f^+, O_f^-)} \end{pmatrix} \end{array}$$

A given element $O_{o,d}(t)$ of $O(t)$ refers to an elementary or composed path ([o, d]) from an origin o to a destination d. $O_{o,d}(t)$ may take one of the following values:

$$O_{(o,d)}(t)=\begin{cases} X: \text{if according to the road law, } [o,d] \text{ could not be traversed in this direction;} \\ *: \text{if there is no offer considering this route (i.e. no vehicule passing by } [o,d]); \\ \text{The set of parameters } (Pl^{j}_{o,d}, De^{j}_{o,d}, A^{j}_{o,d}) \text{ of each vehicle } V_j \text{ serving the route } [o,d] \end{cases}$$

When at least one offer exists on a given route [o, d], related matrix element $O_{(o,\,d)}$ (t) contains several information on the concerned offer(s) for each vehicle V_j serving it:

- $Pl^{j}_{o,d}$: The number of places available within V_j on [o, d]. This parameter could differ from a partial route to another with reference to passengers' deposit and / or pickup at intermediate destinations,
- $[De,A]^{j}_{o,d}(t)$: indicates V_j's Departure time at the earliest from *o* and its Arrival time at the latest on *d* that are calculated as in what follows:
  For each two successive route's segments [h,o] and [o,d] / $[[h,o],[o,d]] \subseteq O_{V_j}(t)$:

$$\begin{cases} De^{j}_{o,d}(t)=De^{j}_{h,o}(t)+TT(h,o,t) \\ A^{j}_{o,d}(t)=A^{j}_{h,o}(t)+TT(h,o,t)+TDT(h,o) \end{cases}$$

Cars' itineraries are composed of several sections and different vehicles may have in common some partial routes or simply intersection points. Each node of $G(t)$ could then have one or more successors referring to potential intermediate or final destinations. This helps distinguish two main sets for each node $x \in N(t)$:

1. The set of successors denoted $N_x^+(t)$ defined as:

$$N_x^+(t)=\{v \in N(t) / (x,v) \in A(t)\}$$

2. The set of predecessors denoted $N_x^-(t)$ defined as:

$$N_x^-(t)=\{v \in N(t) / (v,x) \in A(t)\}$$

Cars' global itineraries are then modeled as n successors' chain, $\forall j \in (1..m)$:

$$IT_{V_j}(t)=(v_0,v_1,...,v_n) \text{ with } \begin{cases} v_k \in N(t), \forall k \in [0,n] \\ v_0 = O^{+}_{V_j}, v_n = O^{-}_{V_j}, \bigcup_{q=1}^{n-1} v_q = ID_{V_j}(t) \\ \forall v_k, v_{k+1} \in IT_{V_j}(t), (v_k, v_{k+1}) \in A(t) \end{cases}$$

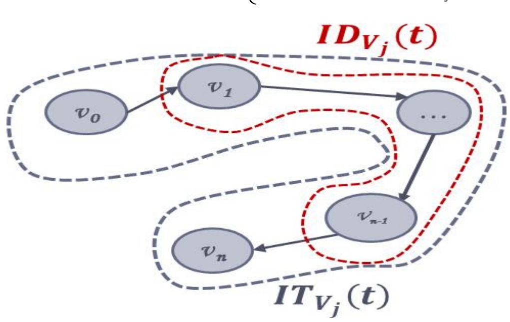


FIGURE 7: A global itinerary composed of several sections.

and so as a composed route of a finite number of Partial Itineraries (*PI*):

$$IT_{V_j}(t) = \bigcup_{s=1}^{n-1} PI_s^j = \{[O_{V_j}^+, ID_1],[ID_1, ID_2],...,[ID_{q-1}, ID_q],...,[ID_z, O_{V_j}^-]\}$$

Where PI_s^j is the partial itinerary s of the vehicle V_j. Two cars could share itineraries intersection without having any common partial route but can also share Partial Itineraries as well. Shared PIs could be served by different vehicles at different times or (almost) simultaneously. As the system may process users' requests generating recomposed solutions (Figure 8) with limited number of transfers if no complete solution exists, responses provided to users in this case are then defined as:

$$IT_{U_i}(t) = \bigcup_{j=1}^{v}\bigcup_{s=1}^{n-1} PI_{s,i}^j$$

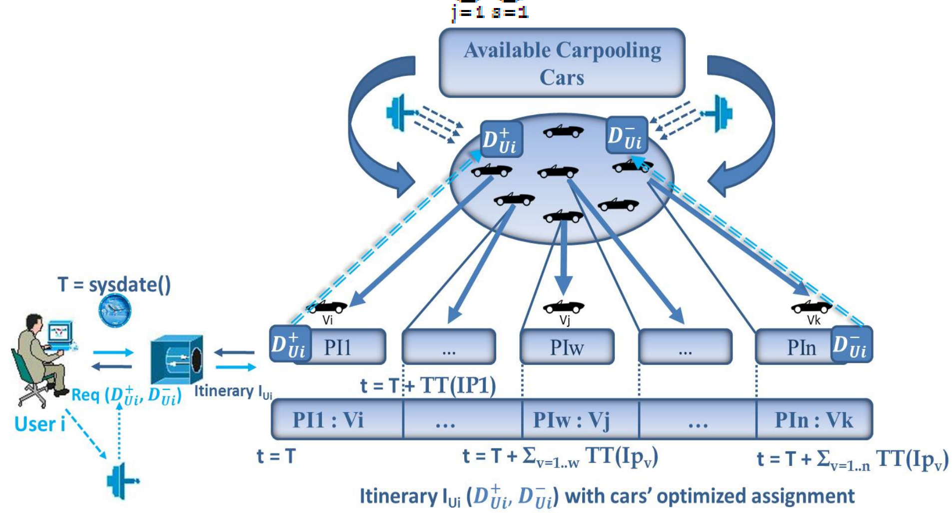


FIGURE 8: A composed optimized solution.

Here $PI_{s,i}^j$ refers to the vehicle V_j that optimizes the Trip Duration on the Partial Itinerary s $(PI_{s,i})$ composing the global solution provided to user i. To be considered as a potential feasible solution on $(PI_{s,i})$, V_j must firstly satisfy some constraints:

- $Pl_s^j \geq P_{U_i}$: V_j must have enough available places Pl_s^j on partial itinerary s according to the number of places asked by user i.
- $PI_{s,i} \in IT_{V_j}$: V_j serves the section s $(PI_{s,i})$ in the correct order from its origin $(PI_{s,i}^+)$ to its destination $(PI_{s,i}^-)$.
- $De_{PI_{1,i}^j} \geq De_{U_i}$: V_j could be assigned to the first leg of the given solution to user U_i if and only if its departure time at the earliest from the concerned route's origin is greater or equal to the one specified by the user.
- $De_{PI_{s,i}^j} \geq A_{PI_{s-1,i}^v}, \forall s \in (1..n-1)$: for each two successive partial itineraries, V_j is considered as a solution on $PI_{s,i}$ only if its departure time at the earliest at its origin $(PI_{s,i}^+)$ is later or equal to the arrival time of the vehicle V_v on the previous

leg destination $(PI^{-}_{s-1,i} = PI^{+}_{s,i})$.

- $A_{PI^{j}_{n-1,i}} \leq A_{U_i}$, the vehicle V_j serving the last route's section of $IT_{U_i}(t)$ must reach the final destination $(d^{-}_{U_i})$ at A_{U_i} at the latest.

Based on this and on the optimization criteria considered, the final decision considering the optimized solution for each partial route (s) is taken on the base of an aggregative fitness function optimizing the trip duration on the considered route:

$$TD^{*}_{s} = Min_{j \in V_s}(TD_{s,j} = w_{1,z}W_{t_{j,s}} + w_{2,z}T_{t_{j,s}})$$

Where V_s is the set of vehicles serving the route s while satisfying the whole set of constraints mentioned above.

The whole global solution corresponds to the one optimizing the global Trip Duration of $IT_{U_i}(t)$:

$$TD^{*}_{U_i} = \sum_{s=1}^{n-1} Min_{j \in V_s}(TD_{s,j}) = \sum_{s=1}^{n-1} Min_{j \in V_s}(w_{1,z}W_{t_{j,s}} + w_{2,z}T_{t_{j,s}})$$

$w_{1,z}$ and $w_{2,z}$ computation details are given in what follows.

A Subdivision Principle to Establish a Physical Distributed Architecture

To optimize execution properties and response delay, requests' acquisition is performed in parallel and then processed simultaneously. According to DOMARTiC's formal specifications, a dynamic graph is used to represent the set of requests and offers. Based on this graph, a distributed architecture is established through a subdivision process. This helped to convey the idea of setting up an optimized decentralized process performing dynamic vehicles' assignment. Hence, we are mainly involved in the way to decompose the whole process into a finite number of less complex tasks executed in parallel. Trying to translate that idea, the first step was to break the geographical served network into several areas of limited surfaces and having the same circular shape and dimensions, created areas have common characteristics:

- They refer to geographical zones with limited perimeters,
- Zones may intersect and have common parts with each other,
- Zones' shape is the same for all the established areas. They are presented as circles which centers are dynamically determined,
- Zones are real time constructed according to the instantly received demands and the available offers at the considered time t ...

The subdivision process adopted [14] determines sets of neighbors coordinates (i.e. nodes of G(t) that are close to each other with respect to the fixed zones' diameter) and establishes several areas. Each one of the created areas corresponds to a given set of neighbors and responds to the stated characteristics. In this decomposing process, two main steps are considered:

a) Step 1: Whereby Primary Zones (*PZ*) are established considering only passengers' requests specifications (i.e. asked origins and destinations) on the base of which groups of neighbors coordinates are determined. A *PZ* is created for each one of them.
b) Step 2: Including drivers' moving specifications and their moving itineraries, the same principle is followed to set up Intermediate Zones (*IZ*) taking into account the firstly established ones (i.e. *PZ*).

This decomposing principle is the base of the decentralized process where optimized responses management is performed in a distributed way over the several created zones. Based on this and as we consider a fitness function aggregating W_t and T_t, their weights differ from one zone to another and are calculated properly to each zone z of the considered areas (z є Z). W_t and T_t weights (i.e. $w_{1,z}$, $w_{2,z}$) are calculated according to the Proportion of Positive Delay (PPD_z):

$$PPD_z(t) = VPD_z[t - \theta, t] / TN[t - \theta, t]$$

This parameter reflects, more faithfully than any other, users' waiting and traveling time within each zone at a given time *t* and helps estimating trips' global duration based on prior knowledge observed on $\Delta t = [t - \theta, t]$ (i.e. the number of cars that started their journeys in z later than firstly indicated (VPD_z: Vehicle having Positive Delay) compared with the Total Number of cars (TN).
In fact, PPD_z shows the impact of different factors (e.g. traffic congestion, disturbances …) observed during an elapsed period of time θ in a zone z and that might directly or not affect the traveling duration. Thus, based on this concept, ω_1 and ω_2 differ from one zone to another and may change over time. Their respective formulas are defined as:

$$\omega_{1,z}(t) = PPD_z(t)$$
$$\omega_{2,z}(t) = 1 - PPDZi(t)$$

The established architecture is then presented as a distributed dynamic graph that is largely appropriate to parallel requests processing jointly considering the multi-agent concept [13].

3.4. Setting up a Distributed Software Architecture Based on the Multi-Agent Concept

Multi-Agent Systems (MAS) are composed of several entities that can be seen as interacting intelligent agents with parallel operations [15]. These agents act in a specific environment according to a certain organization and follow pre-established communication rules to ensure a coherent global process. MAS can be used to solve problems with high complexity. Thus, this concept is perfectly well adapted to represent carpool community where different agents must coexist and cooperate to provide a real time carpooling service as satisfactory as possible.

Review of Existent Multi-agent Systems

MAS have known a great success in the analysis and description of traffic systems increasing traffic components autonomy and facilitating the integration of several frameworks [16, 17]. In classical carpooling systems [18], users are represented by software agents. A special one called super-agent is responsible for managing users' requests searching for possible matchings, but the final choice is yet done by the concerned users and decision is always made by them.
Some mobile-based multiagent applications have been proposed such as MobiAgent [19][7] that allows users to access various services (e.g. web search, remote applications control …) using their mobile phones or PDAs. A Personal Agent (PA) is created in a centralized server for each user sending a request for a specific service. The latter get disconnected from the network and is then notified by the PA of the results it has found. Another System called Andiamo [16] suits for the same principle proposing an agent-based framework using the ToothAgent architecture [20].
Existent ride sharing systems mainly tackle the problem of accessing the system through mobile phones or PDA but remain limited since they are very restricted to the only task of searching for possible matching according to users' moving needs without allowing any flexibility. Thus, automated requests management performed by super-agents was their only important contribution to ameliorate the carpooling service overriding real time and optimization aspects.

The Multi-agent Concept Within DOMARTiC

As stated above, processing dynamic carpooling users' requests is of combinatorial complexity. Besides, the network's modeling previously proposed is suitable for a distributed architecture. This comes to confirm the choice of adopting the multi-agent concept to efficiently deal with dynamic carpooling service. Furthermore, the multi-agent concept is yet more useful as we consider an optimization issue.

Several entities referring, among others, to users' and cars' assistants evolve within our system and are able to communicate and exchange information. Users of our system are then provided with a communication support so that they can instantly interact and communicate with each other and with the carpooling server that is also made up of several entities.

[7] http://www.docstoc.com/docs/58249789/Genghis---A-Multiagent-Carpooling-System

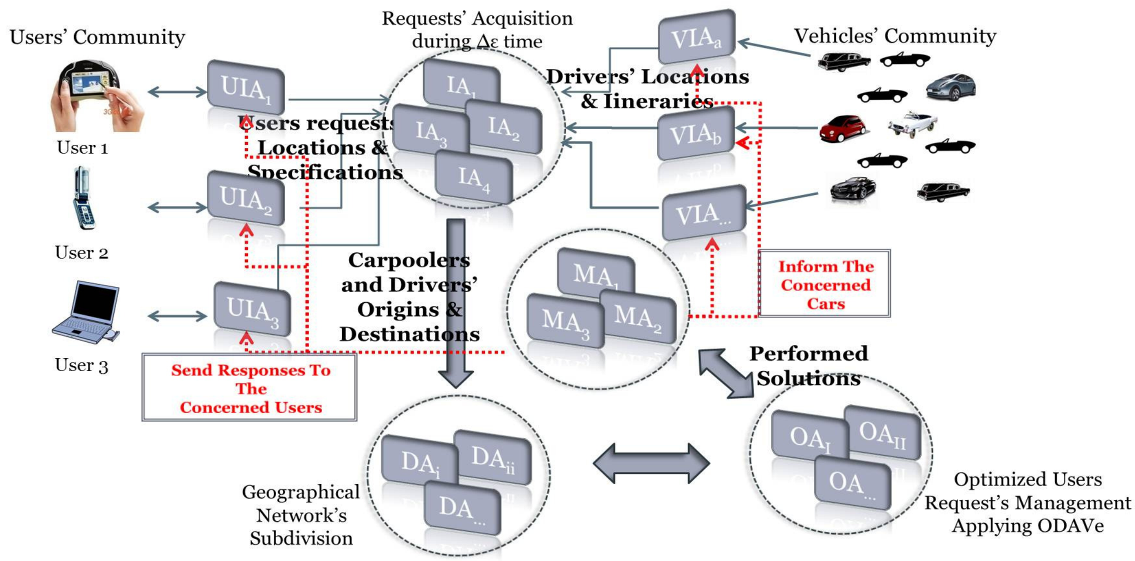


FIGURE 9: Agent modeling within DOMARTiC.

Thanks to this concept, DOMARTiC is provided with an interactive support, enabling the involved entities to exchange information and ensure a coherent process as it is show in Figure 10.

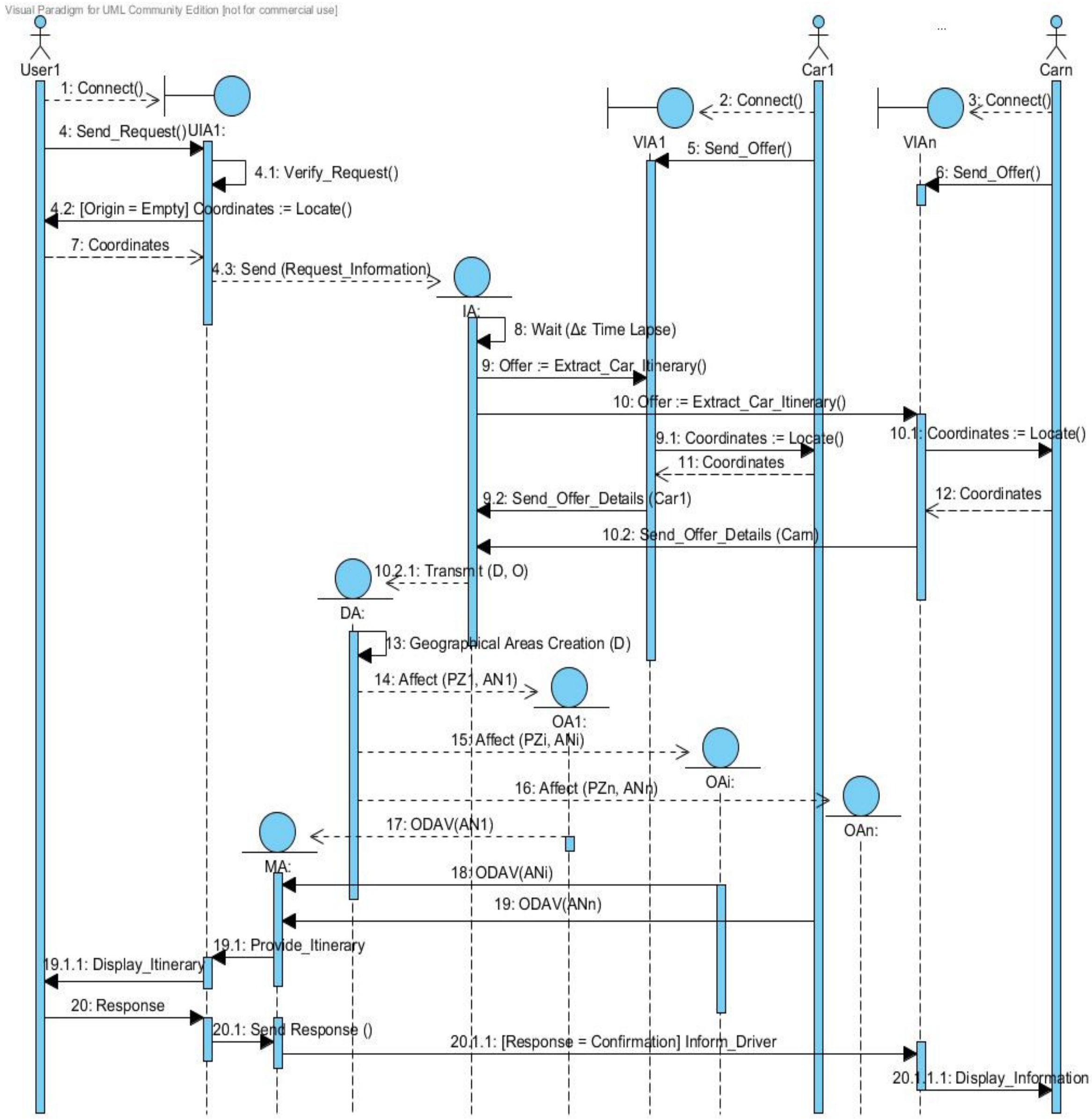


FIGURE 10: Messages exchange to perform DOMARTiC's process.

Agents' functioning:

In this section, entities involved in our system are described with the main features they are responsible for:

User Interface Agent (UIA): created for each user connected to the system and is responsible for receiving its demand and transmits it to the Information Agent (IA). After the optimized process is performed, it provides the user with the generated response,

Vehicle Interface Agent (VIA): ensures driver's exchange with the involved agents (i.e. IA respectively the MA) transmitting the offer it proposes or receiving notifications of pickup and deposit addresses.

Information Agent (IA): must provide the Decomposing Agent with the necessary information on users' requests and the itineraries (i.e. drivers' offers) that might be suitable for those requests. For this purpose, it should firstly perform requests reception during $\Delta\varepsilon$ time, an example of received requests at time t = 9h10 is given in table 2.

User's Identifier	Request's Origin	Request's Destination	Number of Persons	De	A
U_1	U_1^+	U_1^-	1	9h10	10h40
U_2	U_2^+	U_2^-	2	9h10	10h00
U_3	U_3^+	U_3^-	1	9h10	10h15
U_4	U_4^+	U_4^-	2	9h10	10h10
U_5	U_5^+	U_5^-	2	9h13	10h00
U_6	U_6^+	U_6^-	4	9h10	10h30

TABLE 2: Users' requests received at time *t* (D(t))

Secondly, the IA asks for available offers' specifications (table 3) sending a specific message (Request_For_Offer) to each operating VIA. This is based on the concept of agents' coalition to ensure coherence of the responses provided.

Vehicle's Identifier	Offer's Origin	Offer's Destination	Number of Places	ID	De	A
C_1	C_1^+	C_1^-	1	ID_{C_1}	_	10h15
C_2	C_2^+	C_2^-	3	ID_{C_2}	_	13h00
C_3	C_3^+	C_3^-	4	ID_{C_3}	_	11h00
C_4	C_4^+	C_4^-	3	ID_{C_4}	_	12h15
C_5	C_5^+	C_5^-	7	ID_{C_5}	_	11h15
C_6	C_6^+	C_6^-	8	ID_{C_6}	_	12h10
C_7	C_7^+	C_7^-	2	ID_{C_7}	_	12h00
C_8	C_8^+	C_8^-	1	ID_{C_8}	_	9h45
C_9	C_9^+	C_9^-	3	ID_{C_9}	_	13h15
C_{10}	C_{10}^+	C_{10}^-	4	$ID_{C_{10}}$	_	10h30

TABLE 3: Cars' offers available at time *t* (O(t))

Decomposing agent (DA): Having received the required information (i.e. Requests and Offers' specifications) from the IA, the DA is responsible for performing the previously described subdivision principle and according to which it should determine nodes close to each other and create a zone for each set of neighbors and an optimizing Agent for each established zone. Based on available information (table 2 and Table 3), Figure 11 illustrates a view of the served network on real map here limited to the only geographical area including the whole set of offers and demands. In this Figure, users' dispersion through the considered real map are represented according to their location (i.e. current coordinates) captured through GPS tools. Addresses to go to (i.e. final destinations of users and drivers) are also represented.

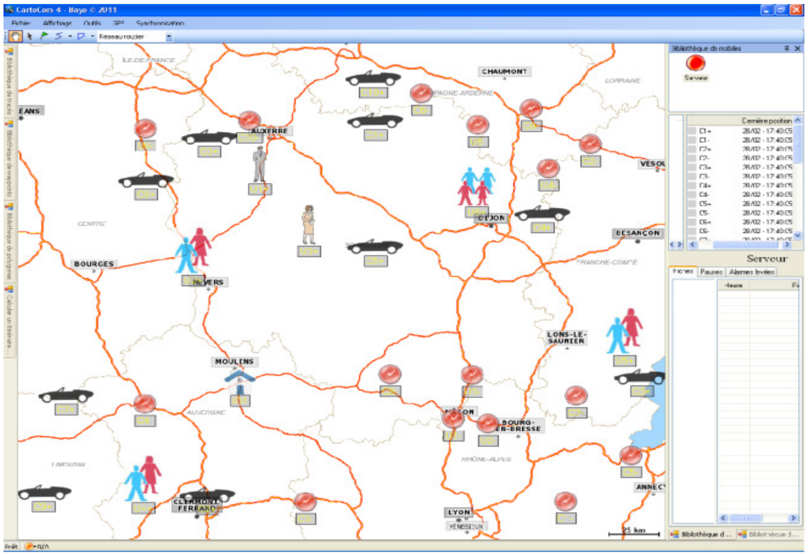


FIGURE 11: Offers and demands' specifications across the geographical network.

To establish such view, we have used automatic geolocation software called Cartocom. The latter integrates world real maps and is based on GPS tools to calculate itineraries based on the information inserted. An example of car itinerary's generation according to its origin, final and intermediate destinations is given by Figure 12.

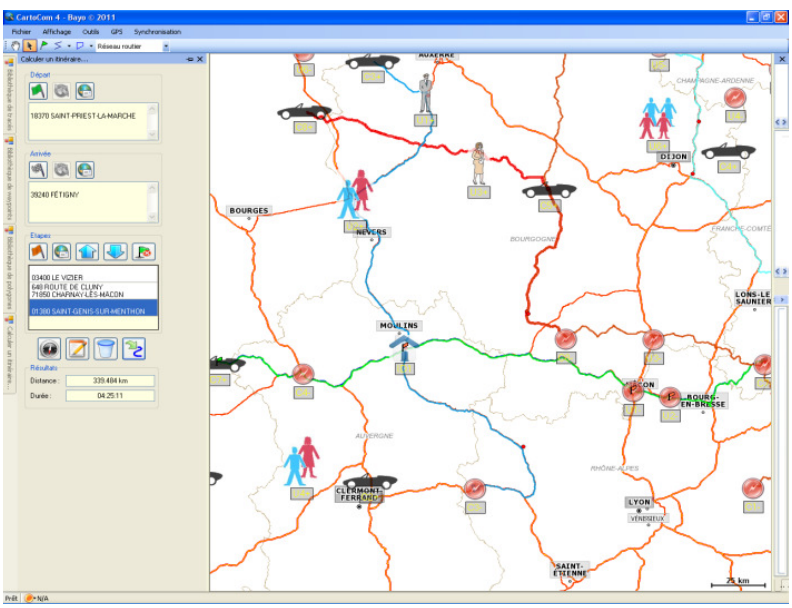


FIGURE 12: C_7' itinerary's view on real maps.

Figure 13 shows how users and cars are spread with their respective itineraries over the considered area, here French regions, as the given example has been tested in France.

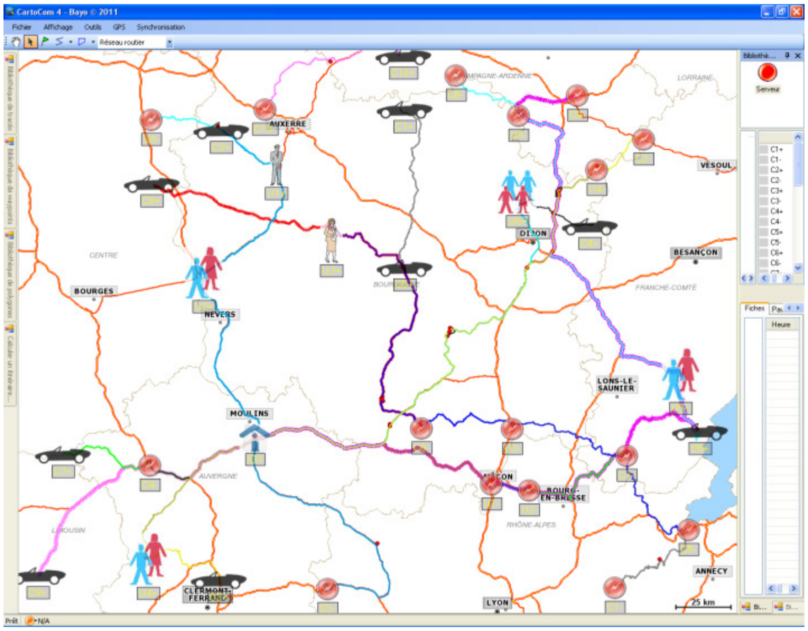


FIGURE 13: Vehicles' and users' scattering over the network.

Based on the established itineraries, a graph modeling is realized defining automated data structures that contain the available information according to the previously defined formalism. Figure 14 illustrates an automated translation of the real network's representation observed in Figure 13.

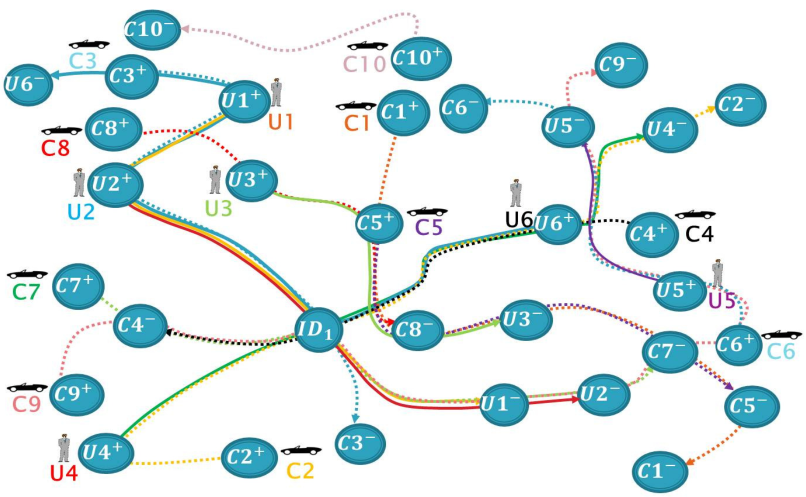


FIGURE 14: Graph modeling of users' requests and cars' offers.

The several steps involved within the decomposition process are then carried out on the established model. Consequently, several sets of neighbors are constructed leading to many distinct zones spread over the network as shown in Figure 15.

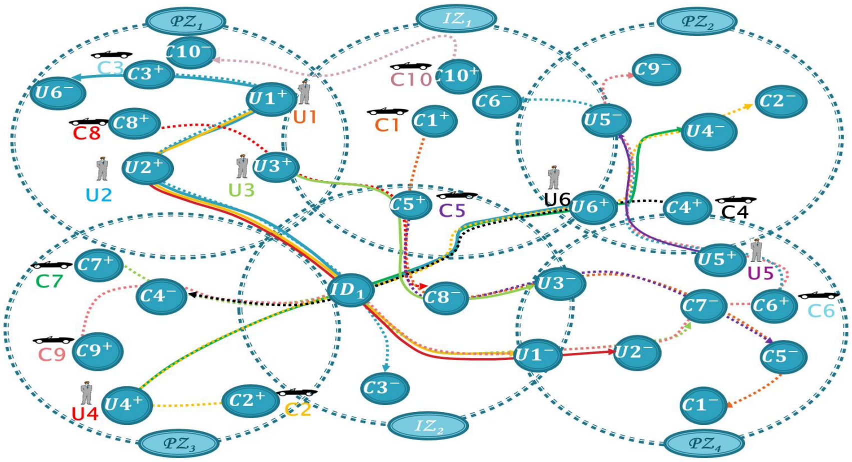


FIGURE 15: Zones established according to the decomposition process.

Optimizing Agent: locally processes optimized algorithms to search for convenient and optimized cars allocation. The optimized responses generation is performed in parallel, in a decentralized way, according to the distributed physical architecture set up through the subdivision process. In fact, an Optimizing Agent is created for each area established by the decomposition process and locally processes users' requests involved within its boundaries performing Algorithm 1. Each OA performs the implemented algorithm processing Optimized and Distributed Assignment of Vehicles to users (ODAVe) in order to find out the best cars' allocation regarding the global traveling duration while ensuring the continuity of the journey (i.e. providing final optimized and complete solution). Before optimization process begins, each zone OA_p determines the optimization's parameters $w_{1,p}$ and $w_{2,p}$.

Algorithm 1 is performed by every Optimizing Agent OA_p responsible for locally processing optimized assignment of vehicles to users requests ($D_{OA_p}(t)$) which specifications (i.e. origins) are included within the zone Z_p it represents. Thus, in the beginning of the process, Optimizing Agents (OAs) representing Primary Zones that include users' requests' origins are firstly launched. In fact, these OAs represent the only zones including users' requests that could be performed at the process first iteration.

However, in later iterations, many Optimizing Agents, may them represent either Primary or Intermediate zones, could be involved within the running process since they could have received requests from other OAs. Indeed, as shown in Algorithm 1 which is performed by each OA receiving a request, the way to process requests is the same for each partial or global initial request:

At first, OA_p has to search for Optimized Global Itinerary (i.e. OGI) responding to the whole asked trip and fitting the previously established constraints (i.e. De, A, number of places ...) (Verify Constraints(...)).

As shown in Algorithm 2, each optimizing agent begins its process searching for possibilities (i.e. vehicles' offers) that may be assigned to a given request. A set of Potential Solutions (PS) is then filled with the extracted feasible solutions. Each potential solution is a data structure representing a given possibility related to a vehicle's offer on a given route. Thus, each single PS is characterized by trip's specifications (i.e. origin, destination, Vehicle Identifier, the number of available places within it, the Global Duration (GD), the departure time at the earliest (De) of the considered vehicle at the given origin, its arrival time (A) at the specified destination, and the related fitness value). Thereafter, the final solution is chosen as the optimal one (PS_o) with regards to the previously stated objective function (i.e. Fitness value for each considered possibility of a specific vehicle).

Data: $D_{OA_p}(t)$: set of Demands included within the represented zone Z_p incumbent on OA_p; O(t): set of Offers; TDT: Tolerable Delay Threshold; TAT: Traveling Average Time; time system t;
Results: The set of Solutions S for the considered requests $(D_{OA_p}(t))$
Initialization: Solution S $\leftarrow \emptyset$;

ω_{1_p}, ω_{2_p} $\leftarrow$ Calculate Optimization Parameters;
For Each Demand D_i **such that** $D_i \in D_{OA_p}(t)$ **do**
$S_i \leftarrow$ OGI (D_i);
If ($S_i = \emptyset$) **then**
For Each Offer O_{V_j} **such that** $O_{V_j} \in O(t)$ **do**
IN $\leftarrow$ IN $\cup$ Intersection (D_i, O_{V_j})
End for
Repeat
NN $\leftarrow$ Nearest_Node (IN, D_i^-)
IN $\leftarrow$ IN \ NN
Set Request $D_{i_1} \leftarrow (D_i^+, NN, P_{D_i}, De_{D_i}, EAT(D_i^+, NN, t))$
$S_{i_1} \leftarrow$ OGI (D_{i_1})
$Z_c \leftarrow$ Containing_Zone (NN) $\{NN \in Z_c\}$
Set Request $D_{i_2} \leftarrow (NN, D_i^-, P_{D_i}, EAT(D_i^+, NN, t), A_{D_i})$
Send Request D_{i_2} to OA_c {OA_c is the Optimizing Agent responsible for performing requests which origins belong to Z_c perimeter}
$S_{i_2} \leftarrow$ Receive Response From OA_c {OA_c performs ODAVe on D_{i_2} and sends partial solutions S_{i_2} to OA_p}
If $(S_{i_2} = \emptyset)$ **then**
$S_i \leftarrow \emptyset$
Else
$S_i \leftarrow S_{i_1} \oplus S_{i_2}$
End if
Until $S_i \neq \emptyset$ Or $IN \neq \emptyset$
End If
$S \leftarrow S \cup S_i$
End for
MA $\leftarrow$ Select a Merging Agent ()
Send S to MA

ALGORITHM 1: ODAV : Optimized Distributed and dynamic Assignment of Vehicles to users.

Secondly, if no car serves the whole considered route or could not fulfill the considered user's expectations in terms of time of departure or/and time of arrival or/and number of places available, etc. OA_p searches for possible itineraries combinations… For this purpose, it must:
Determine a set of Intersection Nodes (IN). Each element of this set corresponds to the closest node served by a given vehicle to the asked final destination. For this purpose, the whole set of existing drivers' offers is considered. IN is then established based on a comparison of the itineraries of each available car with the asked itinerary determining their intersection.
Extract the most approximate served node (i.e. NN : the Nearest Node to the final destination D_i^-),
Decompose the initial request into two distinct demands. The first one is performed by the considered operating OA (i.e. OA_p),
Calculate the necessary parameters (i.e. De, A, ...) characterizing the second partial request,

Determine the zone Z_c containing NN and the corresponding Optimizing Agent (OA_c) over which the whole process (i.e. Algorithm 1) is relaunched dealing with the second partial demand,
Wait for the corresponding solution. The latter must be valid (i.e. not empty) so that the first agent can recompose a whole global itinerary,
Otherwise, step 2 is performed until the received solution is valid as there remain possibilities to test.
According to the previously established distributed architecture and on the base of the given example, 5 OAs have been created to ensure the optimized requests processing. The whole set of solutions provided by the end of the process is illustrated in Figure 16.

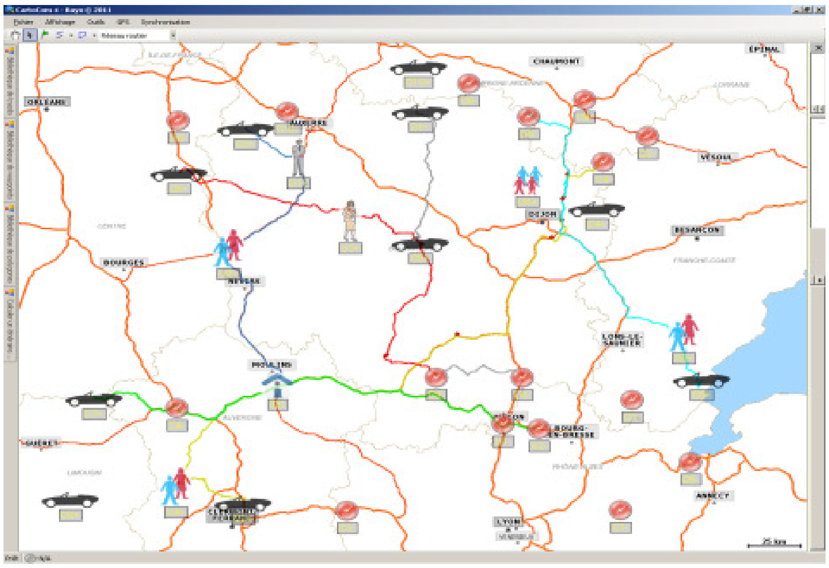


FIGURE 16: Itineraries details within solutions provided to users according to the assigned vehicles

Data: D_i : a specific demand included within the represented zone Z_p incumbent on OA_p;
O (t): set of Offers; time system t; ω_{1_p}, ω_{2_p}
Results: S_i: the global optimized solution performed for the considered request D_i
Initialization: Solution $S_i \leftarrow \emptyset$;

1. **For Each** Offer O_{V_j} such that $O_{V_j} \in O(t)$ **do**
2. **If** $D_i^+ \in IT_{V_j}(t)$ **then**
3. **If** $D_i^- \in IT_{V_j}(t)$ **then**
4. Calculate $(W_t, T_t, GD, De, A, Fitness)$
5. **If** Verify Constraints (D_i, Pl_{V_j}, De, A) = **True then**
6. PS $\leftarrow$ PS $\cup (D_i^+, D_i^-, V_j, Pl_{V_j}, GD, De, A, Fitness)$
7. **End if**
8. **End if**
9. **End if**
10. **End for**
11. $S_i \leftarrow$ Choose PS_o from PS with minimum Fitness
12. **Return** S_i

ALGORITHM 2: Optimized Global Itinerary (OGI).

Applying ODAV:
Here are some results of how the adopted optimizing algorithms are applied over the involved entities (OA) according to the previously given example. Different values of the TAT are represented in Table 4:

Departure Time	Weather	Type of the Road	Traveling Average Time (per 100 km)
Nomal Period	Rain	Highway	Speed=100km/h TAT=75 min
Normal Period	Rain	Secondary Road	Speed=70km/h TAT=120 min
Rush Hour	Rain	Highway	TAT=100 min
Rush Hour	Rain	Secondary Road	TAT=150 min
Normal Period	Normal	Highway	TAT=67 min
Normal Period	Normal	Secondary Road	TAT=86 min
…	…	…	…

TABLE 4: Given information about the Traveling Average Time (TAT)

According to the subdivision process previously performed on the given example, four Optimizing Agents are firstly lunched at the same time dealing with users' requests located within the determined Primary Zones. Then, if partial requests are established with intermediate origins situated in Intermediate Zones, the corresponding agents begin their process at the first request received.
Some computational details are shown through Figures 17, 18, and 19 showing computing operations performed respectively by OA_3, OA_4 and OA_2.

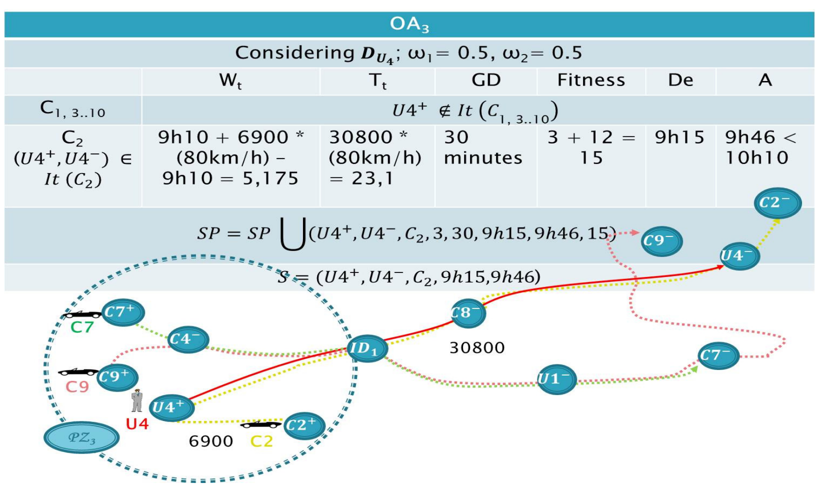
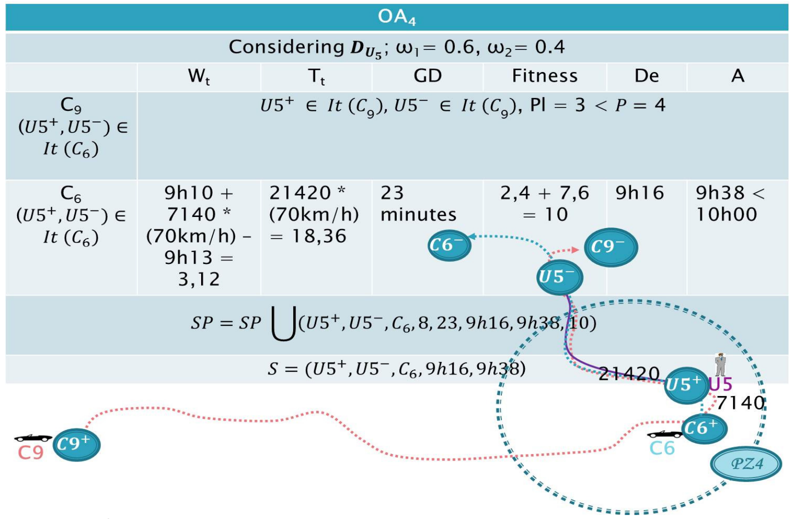


FIGURE 17: Searching for optimized vehicle assignment matching U_4's request appropriately to the specified parameters

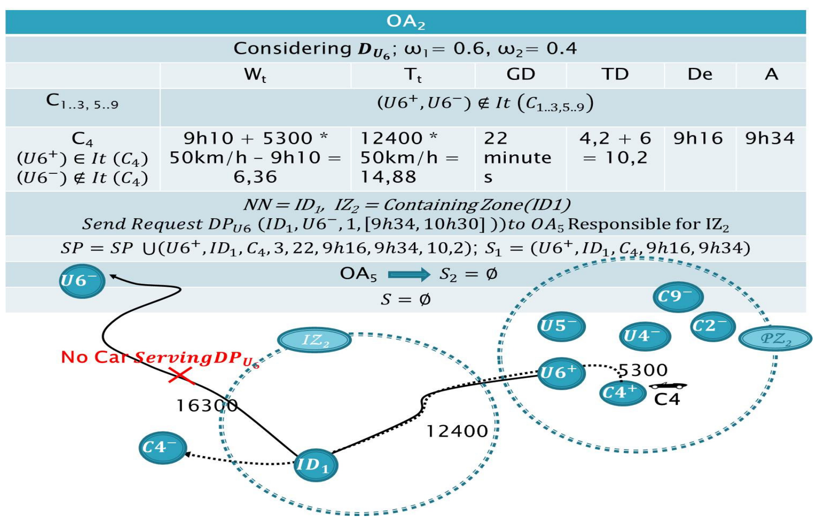
Figure 17 illustrates the operational process performed by OA_3 searching for an optimal journey responding to the only request involved within the zone it is responsible for (PZ_3). It shows a basic scenario where only a car (C_2) may satisfy the asked trip and so presents a feasible but also complete solution. Thus, no optimizing choice is done in this case; C_2 also corresponds to U_4's expectations with regards to the time of Departure at the earliest and Arrival time at the latest.



OA$_4$						
Considering D_{U_5}; $\omega_1 = 0.6$, $\omega_2 = 0.4$						
	W_t	T_t	GD	Fitness	De	A
C_9 $(U5^+, U5^-) \in It\,(C_6)$	$U5^+ \in It\,(C_9)$, $U5^- \in It\,(C_9)$, Pl $= 3 < P = 4$					
C_6 $(U5^+, U5^-) \in It\,(C_6)$	9h10 + 7140 * (70km/h) – 9h13 = 3,12	21420 * (70km/h) = 18,36	23 minutes	2,4 + 7,6 = 10	9h16	9h38 < 10h00
$SP = SP \bigcup (U5^+, U5^-, C_6, 8, 23, 9h16, 9h38, 10)$						
$S = (U5^+, U5^-, C_6, 9h16, 9h38)$						

FIGURE 18: Searching for optimized adequate vehicle to U_5's request

Figure 18 illustrates the process done by OA$_4$ searching for possible solutions to U_5 and which are finally restricted to only one that fits well to the user's expectations. In fact, C_9 serves the route asked by U_5 but doest not fulfil its demand since it does not provide enough places while C_6 presents a sufficient number of vacant places in addition to satisfying the remaining conditions related to U_5's requirements.



OA$_2$						
Considering D_{U_6}; $\omega_1 = 0.6$, $\omega_2 = 0.4$						
	W_t	T_t	GD	TD	De	A
$C_{1..3,5..9}$	$(U6^+, U6^-) \notin It\,(C_{1..3,5..9})$					
C_4 $(U6^+) \in It\,(C_4)$ $(U6^-) \notin It\,(C_4)$	9h10 + 5300 * 50km/h – 9h10 = 6,36	12400 * 50km/h = 14,88	22 minutes	4,2 + 6 = 10,2	9h16	9h34
$NN = ID_1$, $IZ_2 = ContainingZone(ID1)$ Send Request DP_{U6} $(ID_1, U6^-, 1, [9h34, 10h30]\,))$ to OA_5 Responsible for IZ$_2$						
$SP = SP \cup (U6^+, ID_1, C_4, 3, 22, 9h16, 9h34, 10{,}2)$; $S_1 = (U6^+, ID_1, C_4, 9h16, 9h34)$						
OA$_5$ $\Longrightarrow$ $S_2 = \emptyset$						
$S = \emptyset$						

FIGURE 19: A void solution for U_6's request

As shown in Figure 19, there is no solution (i.e. available car) corresponding to the second part composing the initial request (ID$_1$, $D_{\overline{U_6}}$). In this case, as we mainly seek for complete solutions aiming at performing continuous trips till the asked final destination, no solution is provided to the concerned user (U_6) essentially trying to ensure the security of the passengers and their satisfaction.

In addition to performing optimized requests' management, the adopted algorithms helped profit from the multi-agent concept setting up a distributed process over the various agents evolving within our system without having to suffer the inconveniences it may cause namely important communication flows that may condemn the real time aspect. In this context, as the proposed algorithms optimize the process of possibilities generation, they optimize by the way the use of the communication support in the sense that only concerned agents with the extracted possibilities interact and then avoid extra flows.

4. CONCLUSION & FUTURE WORK

With the rise of communication technologies, it becomes relatively easier to deal with problems requiring short process delays. A negligible solving time is of prior order when having to instantly carry out responses for real time queries. Dynamic carpooling is set in this problems category. Therefore, this problem is given priority in this paper. Indeed, to support sustainable mobility mainly based on carpooling, we focus on processing dynamic users' requests in an optimized way. Thus, a new approach called Distributed Optimized approach based on the Multi-Agent concept for Real Time Carpooling service (DOMARTiC) is proposed. In this approach, we introduce a distributed technical architecture to optimize parallel users' requests' processing considering the principle of Multi-Agent systems. In conjunction with the proposed geographical network decomposition, the multi-agent concept highlights the distributed architecture and helps decomposing complex initial tasks to better perform optimized cars' assignment (efficiently and promptly).

Thanks to the proposed graph modeling and the multi-agent concept, the original problem is then decomposed into multiple less complex tasks helping to perform decentralized parallel process over the established areas. This importantly reduces optimization problem's complexity and so helped to set up a process generating optimized responses within a reasonable runtime.

5. REFERENCES

[1] T. Fraichard. *"Cybercar: l'alternative à la voiture particulière"*. Navigation (Paris), 53(209): 53-74, 2005. URL : http://emotion.inrialpes.fr/bibemotion/2005/Fra05.

[2] R. Clavel. *"Le covoiturage en France et en Europe, Etat des Lieux et Perspectives"*. 88, 2007. CERTU, France.

[3] A. Rocci. *"Les évolutions du rapport à l'automobile"*. Journée Technique ATEC-ITS, regards croisés sur l'offre de véhicules en libre service, 2008.

[4] M. Jean. *"Car sharing, taxi, etc.: To share the car use: in france too ?"* T.E.C. ISSN 0397-6513, 158(12 ref.): 22-29, 2000

[5] K.W. Steininger, C. Vogl, and R. Zettl. *"Car-sharing organizations : The size of the market segment and revealed change in mobility behavior"*. Transport Policy, 3(4): 177-185, 1996. URL: http://ideas.repec.org/a/eee/trapol

[6] C. Morency, M. Trépanier, B. Agard, B. Martin, and J. Quashie. *"Car sharing system: what transaction datasets reveal on users' behaviors"*. Intelligent Transportation Systems Conference, 1(4244-1396): 284-289, 2007.

[7] R. Clavel and P. Legrand. *"Le covoiturage dynamique, Etude préalable avant expérimentation"*. 92. CERTU, 2009. URL : http//www.certu.fr.

[8] H. Zgaya. *"Conception et optimisation distribuée d'un système d'information d'aide à la mobilité urbaine : une approche multi-agent pour la recherche et la composition des services liés au transport"*. PHD Thesis, tel-00160802, version 1, Ecole Centrale, Lille, July 2007.

[9] S. Anily. *"Comments on: Static pickup and delivery problems: a classification scheme and survey"*. Sociedad de Estadística e Investigación Operativa 2007 15: 32-34. Published online: 18 April 2007. URL: http://www.springerlink.com/content/908w031877687407/fulltext.pdf.

[10] G. Berbeglia, J.F. Cordeau and J. Laporte. *“Dynamic Pickup and delivery problems”*. European Journal of Operational Research, volume 202, issue 1, pages: 8-15, 2010.

[11] M. Iori and J.J. Salazar González and D. Vigo, *“An Exact Approach for the Vehicle Routing Problem with Two-Dimensional Loading Constraints”*, Transportation Science 41(2): 253-264, 2007.

[12] MM.Hizem. *"Recherche de chemins dans un graphe à pondération dynamique : application à l'optimisation d'itinéraires dans les réseaux routiers"*. PHD Thesis, tel- 00344958, version 1, Ecole Centrale, Lille, 2008.

[13] M.A. Kamoun. *"Conception d'un système d'information pour l'aide au déplacement multimodal : une approche multi-agents pour la recherche et la composition des itinéraires en ligne"*. PHD Thesis, tel-00142340, version2, Ecole Centrale, Lille, 2007.

[14] M. Sghaier, H. Zgaya, S. Hammadi, C. Tahon. *"ORTiC : A novel Approach towards Optimized Real Time CarPooling with an advanced Network Representation Model on siblings”*. In Proceedings of the 12th LSS symposium, Large Scale Systems : Theory and Applications, Lille, France, 2010.

[15] W. Jiao, and Z. Shi. *“A dynamic architecture for multi-agent systems”*. In Proceedings of the 31st International Conference on Technology of Object-Oriented Language and Systems, 253-260, 1999.

[16] F. Sottini, S. Abdel-Naby and P. Giorgini. *“Andiamo: A Multiagent System to Provide a Mobile-based Rideshare Service”*. University of Trento, DIT. Technical Report 06 -097, 2006.

[17] G., M., B., B., A., H. *“Applications of multi agent systems in traffic and transportation”*. In IEEE Transactions on Software Engineering. 144(1):51-60, 1997.

[18] A.B. Kothari. *“Genghis - a multiagent carpooling system”*. B.Sc. Dissertation work, submitted to the University of Bath, May 11, 2004.

[19] C. Carabelea, M. Berger. *“Agent negotiation in ad-hoc networks”*. In Proceedings of the Ambient Intelligence Workshop at AAMAS’05 Conference, Utrecht, The Netherlands. 5–16, 2005.

[20] B. Volha, G. Paolo and F. Stefano. *“Toothagent: a multi-agent system for virtual communities support”*. In Technical Report DIT-05-064, Informatica e Telecomunicazioni, University of Trento (2005).