A multi-agent approach for the dynamic VRPTW

Mahdi Zargayouna[1,2], Flavien Balbo[1,2], Gérard Scémama[1]

[1] Inrets - Gretia laboratory, French National Institute of Transportation Research and their Security.
Avenue du General Malleret-Joinville,
F-94114 Arcueil Cedex.
[2] CNRS-Lamsade laboratory, University of Paris Dauphine.
Place Marechal de Lattre de Tassigny,
75775 Paris Cedex 16, France.
{zargayou,scemama}@inrets.fr, balbo@lamsade.dauphine.fr

Abstract. In this paper, we propose a multi-agent approach to solve a hard combinatorial problem: the vehicle routing problem with time windows. A multi-agent approach is most relevant when dealing with the dynamic version of the problem, where all the customers are not known before the start of the solving process. We focus on both the interaction process and the computation process. For the interaction process, we rely on the Property-Based Coordination model, enabling to limit the interaction to the only agents that might reach a solution. The overall interaction follows the well-known Contract Net Protocol. Concerning the computation process, we propose a new heuristic, based on the measure of the agents' perception field over their environment, which privileges a solving process focused on the future.

1 Introduction

The deliveries of goods to stores, the routing of school buses, the distribution of newspapers and mail etc. are instantiations of a theoretical problem called the Vehicle Routing Problem with Time Windows (VRPTW). In the VRPTW, situated customers in time and space desire to be visited by a fleet of vehicles, each customer specifying a time window inside which she wants to be visited. The exact solving of the VRPTW is of theoretical interest and cannot handle problems with a reasonable size. Therefore, heuristics and meta-heuristics are developed to find good solutions in a reasonable amount of time. In order to meet the reactivity requirement of operational applications, the most promising category of problems is the dynamic version were all the data needed by the system are not known before the computation starts.

Several researches in the multi-agent community have been done in the transportation domain [3, 13]. Making a strong use of bottom-up paradigm as an approach to system design, makes it easier to comprehend a complex reality, by the reification of the components of the system to manage. In the transportation domain, the objective of many applications is the management of distributed entities. In this context, the multi-agent approach simplifies an approach by analogy.

In this paper, we propose a multi-agent solution to the VRPTW problem based on an environment-centered approach. Recently, the Environment for Multi Agent Systems technical forum group (E4MAS) has produced valuable outputs concerning the poorly exploited concept of the environment, notably in terms of roles, responsibilities, architecture, as well as practical applications, thus opening many challenges in terms of modeling, methodology and engineering [11, 17].

The environment has responsibilities [17] that have been applied successfully in the design of transportation applications. The environment being a shared space for the agents, resources and services, its first responsibility is the structuring of the MAS. The environment modeling is a solution to give a space-time referential to a transportation application. For example, the transportation intelligent information systems or more generally location-based services exploit the information about the location of the mobile users in order to adapt their processing. There is a system which has to manage the relation between the mobile user's location and the local services. In the multi-agent solutions, this task can be achieved by the environment. The environment either centralizes the user agent locations and matches against the available information/services [12] or is distributed and gives a homogeneous support to the service design [7]. Moreover, its privileged intermediary role makes the environment a good candidate to support spatially and temporally decoupled coordination models and therefore it is appropriate for applications taking into account dynamic real environment. For instance, in [16], the environment contains fields that are propagated in the environment in a certain range and used by the agents to organize the task assignment for Automatic Guided Vehicles. The MAS environment becomes the common referential that enables agents to adapt their behavior according to the dynamics of the real environment.

The second environment responsibility is to maintain its own dynamics. Following its structuring responsibility, it can also manage the dynamics of the real transportation environment, ensuring the coherence of the MAS. In [16] for instance, the environment ensures the propagation of the fields. Moreover, the environment can ensure services that are not at the agent level or simplify the agent design. In a traffic light control system [1], the environment, which has a global point of view, gives rewards or penalties to self-interested agents according to their local decision.

Because the environment with its own dynamics can control the shared space, its third responsibility is to define rules for the multi-agent system. In a bus network simulation [10], the main role of the environment is to constrain perceptions and interactions of agents. For transportation applications that have an incomplete knowledge, it simplifies the design of the MAS by a clear separation between the roles of the agents and their organization. In the coordinated monitoring of traffic jams application [6], the environment provides organizations, which dynamically evolves according to the current context.

Because the agents are “users” of the services of the environment and in order to really create a common knowledge, the last responsibility of the environment is to make observable and accessible its own structure.

Based on this last responsibility, the Property based Coordination (PbC) principle [19] promotes an effective separation between an observable description of the abstract and real MAS components and the use by the agents of these descriptions to coordinate them. The multi-agent environment contains the recorded descriptions and supports their processing. Initiated by Linda [2], the data-oriented coordination models have focused on what the agents *do*, not what they *are*. In the absence of agents’ states, the context of the interaction cannot include this information. That means that an agent can not condition the reception of information according to its state and/or the state of other agents. Thanks to the PbC principle, the state of the agents is observable in the environment and can be used for coordination purpose.

Our proposal is a VRPTW system based on a multi-agent environment compliant to the PbC principle. The environment in our application structures the MAS components temporally and spatially, so that the interaction between agents is guided be their perception on it, limiting this way the communications cost. Thanks to the PbC principle, the communication between agents is decoupled in space and time, palliating the loss in information in dynamic environments.

This paper is organized as follows. Section 2 describes the VRPTW problem and the Multi-agent approaches to solve it. In section 3, the coordination model of our MAS for the dynamic VRPTW is exposed. Section 4 presents our proposal, and section 5 evaluates its efficiency. Finally, section 6 concludes the paper.

2 The VRPTW problem

2.1 Overview

In the VRPTW problem, there are customers that have to be assigned to a fleet of vehicles. Each customer is assimilated to a request that is situated in time - a time window - and space - a node of the transportation network - and to which is associated a quantity. The vehicles have a limited capacity and they start and end their trips at a unique depot. This problem can model, for instance, the two following scenarios. The first scenario is a company that manages a fleet of vehicles located initially in a local mall. Customers who desire to go back home late from the mall cannot use the regular transportation services. With a VRPTW system, they can connect to the company server asking for a vehicle to take them home not later than a certain time. The second scenario is a company that has a central depot with a big amount of goods, and a fleet of vehicles. The customers of this company are stores that desire to receive goods from the depot. However, they cannot receive the goods at any time, and they specify an interval of time within which they want to be visited by a vehicle, and a service time necessary to its employees to take the goods out of the vehicle.

The data of the problem are the following. Let $[e_0, l_0]$ be the "scheduling horizon" of the system: for all the requests, the time window has to be comprised between e_0 and l_0. Every customer request c_i is described by : n_i the node of the network concerned by the request, q_i the quantity associated with the request, e_i and l_i the earliest and the latest time in which the vehicle can visit the customer c_i, and s_i the service time of the request, which is the time that the vehicle is obliged to spend in n_i before it can leave. Every vehicle has a maximal capacity κ. Two cost matrices are defined: the matrix of travel times and the matrix of travel distances between every pair of nodes of the network.

Given m vehicles, the exact solving of the VRPTW is to find the optimal set of m routes minimizing the total distance traveled by the vehicles. When using approximate methods, the size of the fleet of vehicles has to be found by the system, and the performance criteria becomes:

1. The number of vehicles used,
2. the total distance traveled by all the vehicles.

Delays, breakdowns, no shows are examples of events that imply a dynamic solving of the VRPTW problem and we can say that a fully static system cannot meet operational settings. The term dynamic VRPTW refers most often to a problem where all the customers are not known when the system starts. This problem is equivalent to the openness problem in MAS, where agents can join and leave the system freely - here the agents are customers and vehicles.

In order to take into account the incomplete knowledge of the data, the most widely used methods are insertion heuristics. Insertion heuristics are greedy methods: a customer is assigned to a vehicle once and this assignment is never reconsidered. The choice of the vehicle in which route the customer will be inserted is based on the increase of the general cost - basically, the traveled distance - resulting from this insertion. The advantage of insertion heuristics is that they provide a solution quickly. However, they are said to be myopic since the future consequence of the assignment is not taken into account in the decision. Indeed, the choice is only based on the traveled distance and do not take into account the future availability of the vehicle. The myopic behavior of the insertion heuristics motivated the use of more elaborate measures that take into account the availability of vehicles to future requests.

2.2 MAS modeling of the VRPTW

Multi-agent approaches for the dynamic VRPTW rely most generally on insertion heuristics (see e.g. [8] for a MAS modeling of a static VRPTW). Dial [4] proposes a MAS where the matching of customers with vehicles is based on an *a priori* geographic clustering. To each vehicle is associated a geographic zone, it is responsible of the customers inside this zone. Vehicles of the same region try to insert the customer at a minimum cost. Fischer *et. al* [5] propose a system composed of two types of agents: companies and trucks. Following the Contract Net

Protocol (CNP) [14], the customer connects on a company, which broadcasts its request to all its vehicles. The vehicles answer if they can insert the customer by specifying their insertion cost. The vehicle with the minimum cost is selected to serve the customer. In order to improve the solutions, the companies exchange customers with high costs or that are not feasible, via a blackboard. In [9], the MAS is composed of Interface agents, Customer agents and Vehicle agents. When a customer reveals to the system, the request of a new customer is broadcasted to all the vehicles of the system. The authors allow a customer to announce himself again after it is assigned to a vehicle in order to find a better deal.

These protocols follow the same process as the insertion methods: for each request, the cost of the insertion is computed for each vehicle and the best vehicle is chosen. Thanks to the MAS paradigm, the computation is distributed since the vehicles locally compute the cost of the insertion. The main result is an improvement of the robustness and flexibility of the computation process. Nevertheless, there is a communication cost that we propose to reduce with the use of a data-oriented coordination model called Property-Based Coordination (PbC) [19].

3 Coordination model of the MAS for the dynamic VRPTW

3.1 The space-time coordination problem

As in [9], we propose three categories of agents: interface agents (IA), customer agents (CA) and vehicle agents (VA). The IA is the interface between a human user and the MAS, it takes into account the request of the user and validates it. The IA checks that the space-time description of the request is correct: the time window is included in $[e_0, l_0]$ and the node exists in the network. If this step succeeds, the IA creates a CA that is the representative of the user in the system. After the request of the customer is treated, the CA will inform its user about the best offer that it received and the corresponding vehicle. The role of the CA is to find the best vehicle according to the user request. To reach its objective, it initiates a CNP with the available VAs. The interaction between a CA and a VA is based on their space-time relation: some vehicles cannot satisfy a request, provided their current plan - composed of other customers. In order to avoid useless communication between CAs and VAs and processing, we propose to limit the communication to the only VAs that are able to satisfy the CA's request, using a data-oriented coordination model.

In the data-oriented coordination models, the reading and the retrieval of data from the shared data space is done by contents. Indeed, an agent specifies a template and the first data matched with it is returned. A specificity of the PbC model is that the data space is the multi-agent environment and the retrieval of a data is assimilated to a perception over this environment. In data-oriented coordination models, agents perceive the data exchanged in the data space, while in the PbC model, the observable state of the agents can be perceived as well. In our problem, the state of the CAs and the state of the VAs can be used in order to limit the communication. For instance, the perception of a request by a VA depends on the state of the VA, e.g. its current capacity, in addition to the space and time location of the request. By defining perception conditions in the environment, the communication is limited, since the requests perceived by the vehicles will be circumscribed to those of a subset of the CAs present in the environment, i.e. those that the vehicle can reach without violating any space-time constraint. Moreover, the relation between a VA and its perception of the environment can be used to propose a new heuristic.

3.2 Coordination model: the PbC model

As said above, in PbC the data space is the MAS environment. Our environment model consists of a set of m entities, $\Omega = \{\omega_1, ..., \omega_m\}$ and a set of k filters, $\mathcal{F} = \{f_1, ..., f_k\}$. An entity ω is related to a component of the MAS (agent, percept, object) and has a description given by observable properties. Let $P = \{P_1, ..., P_n\}$ be the set of the n observable properties of the MAS. An observable property P_i is a function that gives for an entity ω a value that can be used for the interaction, $\forall P_i \in P,\ P_i : \Omega \rightarrow d_i \cup \{unknown, null\}$, with d_i the description domain of P_i, its value set or its type. The value of an entity e for the property P_i is given by $P_i(e)$. The agents are the *active* entities that change the state of the MAS environment by adding/retrieving objects. The agents are not only processes that exchange data via the data space, they are also represented by data describing their current observable states (their properties).

The matching in PbC is based on the notion of a filter. A filter $f(a, o, C)$ is defined as a conjunction of conditions. It describes the agent a interested by the perception, the entity o it wants to perceive, and the context C of the perception i.e. a set of entities (might be empty). A context is the observable state of the environment at a certain moment, e.g. the presence of a certain vehicle in the vicinity of a given node.

Definition 1 (Filter). *Let $A \subset \Omega$ be the subset of agents, let $O \subset \Omega$ be the subset of objects, the filter f is defined as follows, $f : A \times O \times \mathcal{P}(\Omega) \rightarrow \{true, false\}$*
$f(a, o, C) = f_{ag}(a) \wedge f_{pe}(o) \wedge f_{co}(C)$ where:
-$f_{ag}(a) = \wedge_{P_i \in P_{f_{ag}}} [P_i(a) R_{P_i}^{f_{ag}} d_{P_i}^{f_{ag}}]$
-$f_{pe}(o) = \wedge_{P_i \in P_{f_{pe}}} [P_i(o) R_{P_i}^{f_{pe}} d_{P_i}^{f_{pe}}]$
-$f_{co}(C) = \wedge_{c \in C} g_{co}(c)$, with
$g_{co}(c) = \wedge_{P_i \in P_c} [P_i(c)\ R_{P_i}^{c}\ d_{P_i}^{c}]$.

For instance, the filter $f(a, o, \{a_2\}) = [id(a) = \text{“}a_1\text{”}] \wedge [id\text{-}owner(o) = id(a_2)] \wedge [state(a_2) = \text{“}busy\text{”}]$ conditions the perception of an object to the observable state of its owner. The first condition indicates that the filter concerns the perception of an agent having the constant “a_1” as identifier. This agent a_1 perceives the objects which owner is a_2, only if the latter is busy.

4 A MAS based on PbC for the dynamic VRPTW processing

We rely on the PbC model to manage the interactions inside the MAS. The central idea in using the PbC model is to allow VAs to specify their current insertion possibilities by filters in the environment so that they perceive the only CAs they can actually insert in their plan.

4.1 MAS components description

We use PbC to model a system for the VRPTW. In our system, the choice of the properties to be made observable is directed by their use in filters. If a property is not used in any filter, then there is no need to make it observable. The opportunity of this choice is made clear when considered together with the filter in which the properties are used. A CA is described by the following properties (see section 3.1): n, e, l and s. For instance, let a be a CA, $n(a)$ value is the node of the network where the request has been located by the customer. Moreover, the selection of a request by a vehicle is done according to the feasibility of its insertion in its current plan. This constraint implies that the link between the plan of the vehicle and its composing customers has to be observable by the environment. For a CA c, we define the properties *veh* and *succ* representing respectively the VA v to which plan the CA c belongs and the successor of c in the plan of v. Both properties are evaluated to *unknown* if the CA is still searching for a VA to be associated to. These properties, together with the properties e, l are updated according to the result of the solving process, detailed in the next section. In order to express the capacity constraint, we define a property *cap* for the VA representing its capacity. Moreover, each agent in the environment has the property *id* that gives its identity.

The messages that these agents exchange in the CNP are objects in the environment. Each message has two properties: the *sender* and the *receiver*. The values of these properties are the *id* of an agent or *unknown* if the message is not addressed, e.g. a request. For a request object, we define the following observable properties: the node (n), the time window made of a lower bound (e) and an upper bound (l), the service time (s), the quantity (q) and an *id*, which are the characteristics desired by the CA that submitted the request. When a CA puts a request in the environment, it initiates the CNP and gives the information that are needed for the solving process. In order to express the space-time constraint, we define an object Edge. An Edge specifies the travel times between every pair of nodes of the network, it has three properties: the origin node *on*, the destination node *dn* and the cost associated *cost*. This description of the MAS components gives the information that are used in the computation process. In the next sections, we give the execution of the CNP protocol using PbC.

4.2 Vehicle selection

In the CNP, the first step is the search by the initiator of the protocol of the agents that are able to satisfy its request. With PbC, this is done by a CA that puts a request in the environment and by its perception by the related vehicles. This step implies that a filter compares the request description with the current observable state of the MAS. If no VA can satisfy the request, then a new VA is created.

The main constraint of the VRPTW is the following. The insertion of a customer c^* in the plan of a vehicle v^* is feasible if:

- the quantity associated to the request of c^* does not exceed the current capacity of v^*,
- There exists a position in the plan of v^* between two adjacent customers, say c_1 and c_2, so that v^* can visit c_1, then c^* and finally c_2 without violating any time window of the three customers (cf. figure 1).

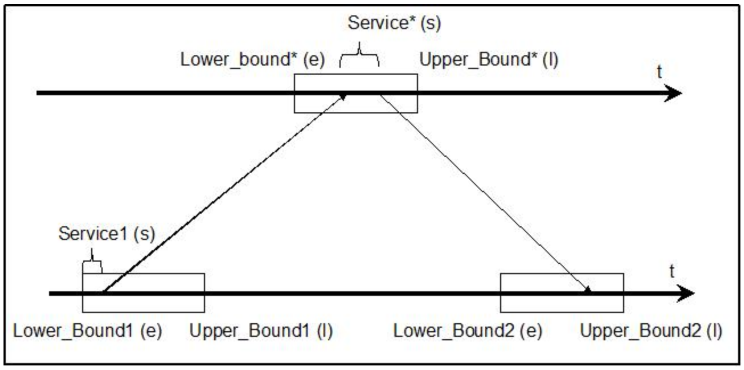

Fig. 1. Feasible insertion

The filter expressing these constraints is complex and is decomposed in two parts $f_{capacity}$ and $f_{space-time}$ in order to simplify the presentation. The capacity constraint is related to the insertion of a customer c^* that submitted a request r^* in the plan of the vehicle v^* , it is defined as follows:

$$f_{capacity}(v^*, r^*, \emptyset) = [q(r^*) \leq cap(v^*)]$$

The VA that inserts a customer updates its current capacity *cap* by withdrawing the quantity associated to the new customer. This way, the VA will henceforth perceive the only requests that respect its capacity constraint.

The space-time constraint related to the insertion of a customer c^* that submitted a request r^* in the plan of the vehicle v^*, between two customers c_1 and c_2 is specified as follows:

$$f_{space-time}(v^*, r^*, \{c_1, c_2, a_1, a_2\}) =$$

$$[succ(c_1) = id(c_2)] \wedge [veh(c_1) = id(v^*)] \wedge$$

$$[n(c_1) = on(a_1)] \wedge [n(r^*) = dn(a_1)] \wedge$$
$$[e(c_1) + cost(a_1) + s(c_1) \leq l(r^*)] \wedge$$
$$[n(r^*) = on(a_2)] \wedge [n(c_2) = dn(a_2)] \wedge$$
$$[Max(e(r^*), e(c_1) + cost(a_1) + s(c_1)) +$$
$$s(r^*) + cost(a_2) \leq l(c_2)]$$

The context of the filter describes the position where the customer can be inserted. It is composed of the two customers c_1 and c_2 between which the customer might be inserted and the two edges a_1 and a_2 linking respectively c_1 to c^* and c^* to c_2. The conditions $[succ(c_1) = id(c_2)]$ and $[veh(c_1) = \text{“}v^*\text{”}]$ identify two adjacent CAs in the plan of the considered VA v^*. The rest of the conditions express the feasibility of the insertion as described above. First, the two edges a_1 and a_2 are identified ($[n(c_1) = on(a_1)] \wedge [n(r^*) = dn(a_1)]$ and $[n(r^*) = on(a_2)] \wedge [n(c_2) = dn(a_2)]$). Second, the feasibility of the insertion of c^* after c_1 ($[e(c_1) + cost(a_1) + s(c_1) \leq l(r^*)]$) and before c_2 ($[Max(e(r^*), e(c_1) + cost(a_1) + s(c_1))) + s(r^*) + cost(a_2) \leq l(c_2)]$) are checked. As a consequence, the filter f_{VRPTW} of a VA that verifies the capacity constraint and the space-time constraint is $f_{VRPTW}(v^*, r^*, \{c_1, c_2, a_1, a_2\}) = f_{capacity}(v^*, r^*, \emptyset) \wedge f_{space-time}(v^*, r^*, \{c_1, c_2, a_1, a_2\})$.

Each time that this filter matches, a VA receives the request description, plus the *id* of the customer after which c^* can be inserted (c_1 in the filter). This allows the VA v^* not to check all the possible insertion positions, since this information is caught by the filter. In order for the filter f_{VRPTW} to allow a new Vehicle v to participate in the interaction, v must have at least two customers in its plan (c_1 and c_2), otherwise, v won't be able to perceive anything. Hence, when created, a VA has two customers d' and d" by default in its plan. These customers do not correspond to human users, but represent the depot. These customers have the following description: $s(d') = s(d") = 0$, $e(d') = e(d") = e_0$, $l(d') = l(d") = l_0$, $q(d') = q(d") = 0$, $n(d') = n(d") = depot$. Thus, for a VA with an empty plan, the insertion of a customer is feasible if the vehicle can leave the depot, visit this customer and go back to the depot without violating its time window.

4.3 The choice criterion

The next part in the CNP protocol, is the computation of a price by each VA that received a request. When using an insertion heuristic, the decision to assign a customer to a vehicle is based on the increase of the cost (the traveled distances). This measure is myopic, because it does not take into account the future, and an insertion decision might imply a greater insertion cost later. The problem we point out is that the solutions generated by any algorithm based strictly on a cost function that do not take into account future requests, tends to generate solutions that are very constrained, i.e. with tight spatiotemporal gap between customers in the plan of the vehicle. As a consequence of the use of this measure, the algorithm leaves fewer future insertion possibilities, and the consideration of new requests might oblige the system to create new vehicles. The intuition of the measure we propose is to focus on the future insertion possibilities. A perception field, that is made of all the positions where it can be in the future and their associated times is associated to every VA. The perception field is maximal for a VA with an empty plan and decreases monotonically with the insertion of new customers in its plan, since a new insertion obliges the VA to sacrifice some space-time positions in order to visit the new customer without violating its time windows.

In order to illustrate the measure, we consider an Euclidean VRPTW. In the Euclidean-VRPTW, the network is a plane, and the travel times are computed with the Euclidean metric, i.e. the travel time between two nodes, say i and j with coordinates (x_i, y_i) and (x_j, y_j), is equal to $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. In this context, a VA v with an empty plan has a perception field described by the figure 2 (the coordinates of the depot are (x_0, y_0)). The set of all possible 3-tuples (x, y, t) that the vehicle can have are those located inside the cone of vertex (x_0, y_0, e_0). The measure associated to the perception field of this VA is the volume of this cone. For every CA inserted in the plan of a VA, the latter updates its perception field and the measure associated to it.

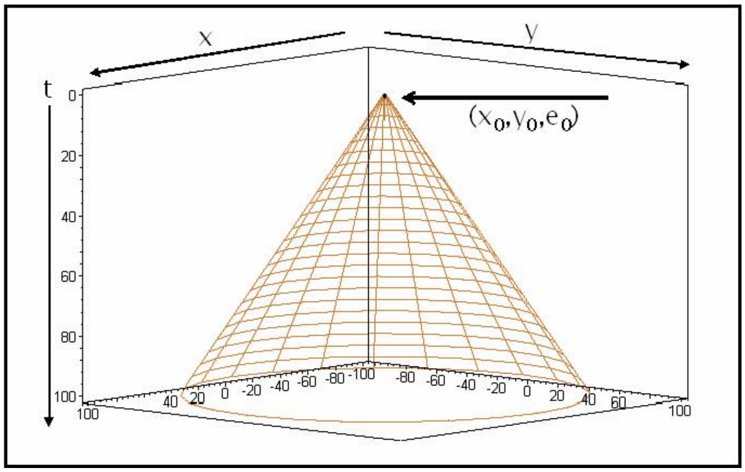

Fig. 2. Initial perception field

A VA receives the only requests that it can satisfy - those located inside the cone -, plus the *id* of the customer after which the insertion can be done. With these data, the VA computes locally the cost of the insertion. Consider a CA c_2 that submitted a request and let us suppose that the VA v is a candidate for its insertion. The VA v has to infer its new perception field if it had to insert c_2. The new perception field is described by the figure 3. The new 3-tuples (x, y, t) accessible to v if it had to insert c_2 are those of the old perception field that are still accessible after the insertion of c_2 (the space inside the bold black line). The

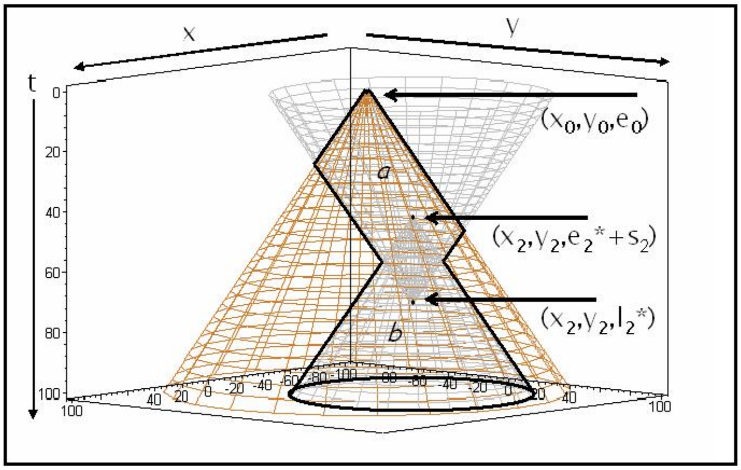

Fig. 3. Perception field after the insertion of c_2

price is computed by the VA v with respect to the insertion of c_2 and is equal to the measure associated to the old perception field minus the new perception field resulting from the insertion of c_2. This price reflects the decrease in the probability of participation of v to the insertion of future requests.

The last part of the CNP is the sending of offers from the VAs that received the request to the CA (c_2 in the previous example). The offers are addressed messages. The perception of addressed messages is done by the following filter :

$$f_{dyadic}(a, m, \emptyset) = [id(a) = receiver(m)]$$

Each concerned VA sends to the CA its price for the request and the new time window of the customer if it had to be inserted in the plan of the VA. The CA chooses among the offers that it receives the one with the lower price and following the CNP, it sends an addressed accept message to the selected VA, and addressed reject messages to the others. This VA informs the customers already inserted in its plan about the consequence of this insertion (some of them will see their time windows narrowed). Each contacted CA updates its related properties (e and l). When the human users are actually visited by a vehicle, they are visited at the time window's lower bound (property e) of their representative CA.

5 Results

Our system was coded in JAVA and executed on a Personal Computer with a Pentium$^{\text{TM}}$IV 3.2 Ghz processor. We ran it with the Solomon's benchmarking problems [15] with three versions: 50, 100 and 200 customers. Three classes of problems are available: C for clustered customers, R for uniformly distributed customers and RC which is a mix of the two. We chose to use the R1 class (with 10 files) since we do not associate any weighting function with customers' location and since we suppose, in the absence of any historical data, that unknown requests may concern any location, following a uniform distribution. Our system starts with no requests known in advance, because we choose to consider each customer independently from its successors. The first indicator we are interested in is related to the interaction process, via the PbC model, materialized by the number of messages received by the VAs. We compare our approach with a broadcast approach, consisting on sending every new request to all the available vehicles. The figure 4 shows that our approach becomes more and more relevant as we consider more and more customers. We see that the number of useless communications is increased when we consider more customers, and the use of PbC becomes more relevant in the presence of more agents in the MAS.

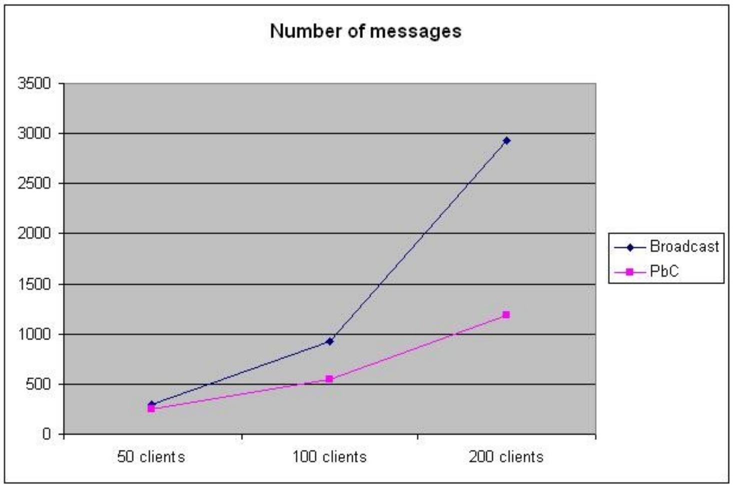

Fig. 4. Number of messages : PbC vs Broadcast

The second indicator is related to the solving process. When the size of the fleet of vehicles is fixed in advance, the central concern with a dynamic VRPTW system is the amount of rejected requests, which should be limited. However, since we create vehicles dynamically when no vehicles can serve a new customer, our system does not reject any request; hence, our central concern becomes the size of this fleet. We have implemented a system which behavior is similar to ours. The only difference is the price computed by a VA, which is equal to the increase of the traveled distance, and not the loss of perception field as we do. The increase of the traveled distance is the traditional measure in insertion heuristics. Table 1 reports the results with files of the class R1 where we consider successively 25, 50, 100 et 200 customers. The results show that the use of our measure mobilizes less vehicles the the traditional measure, and this is the case whatever the number of considered customers. This result validates the intuition of the measure which consists on the maximization of future insertion possibilities for a VA. However, since our measure focuses exclusively on insertion feasibilities, the total distance traveled by all the vehicles with our measure is superior to the distance traveled with the traditional measure. We think that a compromise between the two measures, e.g. a weighted sum of the increase of the distance and the loss of perception fields is able to give better results w.r.t the two criteria.

Δ Distance		
	Number of vehicles	Distance
25	3,4	316,6
50	6	671,2
100	12,1	1601,3
200	21,6	6315,5
Δ Perception Fields		
	Number of vehicles	Distance
25	3,3	347,9
50	5,9	731,5
100	11,9	1774,4
200	21,4	6979,8

Table 1. Experimental results with on class R1 with 50, 100 et 200 customers

6 Conclusion

In this paper, we propose a multi-agent approach for the dynamic VRPTW. We focused our presentation on Vehicle agents and Customer agents, and on their dynamic linking. The use of the PbC model allows us to circumscribe this link to the VAs and CAs which constraints allow for a possible insertion, thanks to filters. The new measure we propose in this paper is based on the perception fields of the agents. Indeed, VAs compute a price corresponding to the insertion of a CA, by computing their loss of future space-time positions. The VA that has the minimum loss is selected, privileging thus future insertion possibilities instead of immediate cost optimization.

In [18], we have proposed a coordination language called Lacios inspired by PbC to unambiguously specify the behavior of the agents of the MAS, and a programming language to implement open MAS in Lacios and run it in Java. We also have proposed a complete solution for a more constrained application than the VRPTW: the Dial-A-Ride problem in which human users desire to be transported from a point of the network to another one inside a time window.

In this paper, we focused on the number of used vehicles criterion. We transformed this global objective - the minimisation of the number of vehicles used - to a local objective - the maximization of the perception fields width. Future works focus on the generalization of the perception fields measure to other distributed applications. Indeed, it would be interesting to assess the shift of a problem objective to a perception fields optimization in the general case.

Concerning the transport applications, we will consider problem versions where the dynamicity concerns changes in travel times as well. The checking of the feasibility of a request under travel times changes will be handled by specific filters. In addition, future experiments will mix the measure of perception field with the traditional increase of travel distance metric.

References

1. Ana L. C. Bazzan. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multiagent Systems*, 10(1):131–164, March 2005.
2. Nicholas Carriero, David Gelernter, and Jerrold Leichter. Distributed data structures in linda. In *POPL'86: Proceedings of the ACM symposium on Principles Of Programming Languages*, pages 236–242, New York, NY (USA), 1986. ACM Press.
3. Paul Davidsson, Larry Henesey, Linda Ramstedt, Johanna Trnquist, and Fredrik Wernstedt. An analysis of agent-based approaches to transport logistics. In *Transportation Research part C: emerging technologies (special issue)*, volume 13(4), pages 255–271. Elsevier, 2005.
4. Robert B. Dial. Autonomous dial-a-ride transit introductory overview. *Transportation Research Part C: Emerging Technologies*, 3(15):261–275, 1995.
5. Klaus Fischer, Norbert Kuhn, and Jrg Mller. Distributed, knowledge-based, reactive scheduling of transportation tasks. In *AIA'94:Proceedings of the Conference on Artificial Intelligence for Applications*, pages 47–53, San Antonio, TX (USA), 1994.
6. Robrecht Haesevoets, Bar Van Eylen, Danny Weyns, and all. Context-driven dynamic organizations applied to coordinated monitoring of traffic jams. In Danny Weyns, Sven Brueckner, and Yves Demazeau, editors, *Proceedings of Workshop Engineering Environment-Mediated Multiagent Systems*, pages 126–143, 2007.
7. Jason I. Hong and James A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *In Proceedings of International Conference on Mobile Systems, Applications, and Services*, pages 177–189, June 2004.
8. Mariem Kefi and Khaled Ghedira. A multi-agent model for the vrptw. In *Proceedings of Urban Transport'04*, Germany, 2004.
9. Robert Kohout and Kutluhan Erol. In-Time agent-based vehicle routing with a stochastic improvement heuristic. In *AAAI'99/IAAI'99: Proceedings of the national conference on Artificial intelligence and the conference on Innovative applications of artificial intelligence*, pages 864–869, Menlo Park, CA (USA), 1999. AAAI Press.
10. David Meignan, Olivier Simonin, and Abderrafiaa Koukam. Adaptive traffic control with reinforcement learning. In *4th Workshop on Agents in Traffic and Transportation (ATT)*, pages 50–56, 2006.

11. Eric Platon, Marco Mamei, Nicolas Sabouret, Shinichi Honiden, and H. Van Dyke Parunak. Mechanisms of the environment for mutli-agent systems, survey and opportunities. *Autonomous Agents and Multi-Agent Systems*, 14:117, 2006.
12. Alexander Repenning and Andri Ioannidou. Mobility agents: guiding and tracking public transportation users. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 127–134, New York, NY, USA, 2006. ACM Press.
13. R. Schleiffer. Intelligent agents in traffic and transportation. In *Transportation Research part C: emerging technologies (special issue)*, volume 10C 5-6, pages 325–527. Elsevier, 2002.
14. Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.
15. Marius M. Solomon. Algorithms for the vehicle routing and scheduling with time window constraints. *Operations Research*, 15:254–265, 1987.
16. Danny Weyns, Nelis Boucké, and Tom Holvoet. Gradient field-based task assignment in an agv transportation system. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 842–849, New York, NY, USA, 2006. ACM.
17. Danny Weyns, H. Van Dyke Parunak, Fabien Michel, Tom Holvoet, and Jacques Ferber. Environments for multiagent systems, state-of-the-art and research challenges. *Lecture Notes in Computer Science Series*, 3374:2–52, 2005.
18. Mahdi Zargayouna. *Modèle et langage de coordination pour les systèmes multi-agents ouverts. Application au problème du transport à la demande*. PhD dissertation, University of Paris-Dauphine, Paris (France), 2007. In french.
19. Mahdi Zargayouna, Julien Saunier Trassy, and Flavien Balbo. Property based coordination. In *AIMSA'06: Artificial Intelligence: Methodology, Systems, Applications*, volume 4183 of *Lecture Notes in Artificial Intelligence*, pages 3–12. Springer-Verlag, 2006.